



Titre: Étude des propriétés des codes convolutionnels récurrents
Title: doublement-orthogonaux

Auteur: Éric Roy
Author:

Date: 2012

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Roy, É. (2012). Étude des propriétés des codes convolutionnels récurrents
Citation: doublement-orthogonaux [Ph.D. thesis, École Polytechnique de Montréal].
PolyPublie. <https://publications.polymtl.ca/1010/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/1010/>
PolyPublie URL:

**Directeurs de
recherche:** David Haccoun, & Christian Cardinal
Advisors:

Programme: Génie Électrique
Program:

UNIVERSITÉ DE MONTRÉAL

ÉTUDE DES PROPRIÉTÉS DES CODES CONVOLUTIONNELS RÉCURSIFS
DOUBLEMENT-ORTHOGONAUX

ÉRIC ROY
DÉPARTEMENT DE GÉNIE ÉLECTRIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION
DU DIPLÔME DE PHILOSOPHIÆ DOCTOR
(GÉNIE ÉLECTRIQUE)
DÉCEMBRE 2012

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée :

ÉTUDE DES PROPRIÉTÉS DES CODES CONVOLUTIONNELS RÉCURSIFS
DOUBLEMENT-ORTHOGONAUX

présentée par : ROY Éric

en vue de l'obtention

du diplôme de : Philosophiæ Doctor

a été dûment acceptée par le jury d'examen constitué de :

M. FRIGON Jean-François, Ph.D., président

M. HACCOUN David, Ph.D., membre et directeur de recherche

M. CARDINAL Christian, Ph.D., membre et codirecteur de recherche

M. HERTZ Alain, Doct. ès Sc., membre

M. CHOUINARD Jean-Yves, Ph.D., membre

À tous mes professeurs . . .

Remerciements

Dans un premier temps, j'aimerais remercier ma seconde famille, qui depuis fort longtemps croit en moi. Messieurs David Haccoun et Christian Cardinal merci pour votre aide financière, sans elle il m'aurait été impossible d'accomplir ce travail, et d'atteindre ce niveau d'excellence. Merci d'avoir cru en mes idées. Merci de m'avoir formé et de m'avoir fait découvrir une communauté de chercheurs extraordinaires. Merci aussi d'avoir partagé cette petite idée qui naquit il y a pratiquement dix ans maintenant. J'espère avoir pris suffisamment soins d'elle pour que les fruits présentés dans ce document puissent être récoltés un jour.

À ma soeur Annabelle, à mon frère Luc et à mes parents Claudette et Jean merci pour vos encouragements.

Un merci particulier à mon ami David Rogers, qui m'a introduit aux études supérieures et qui m'a fait découvrir le monde fascinant des télécommunications il y a quelque treize années déjà. Ta détermination m'a été d'une grande inspiration.

À mes amis du laboratoire, plus particulièrement Wael, Zouheir, Laurent et Gilbert, je suis très heureux d'avoir fait votre connaissance, merci d'avoir comblé le vide du local M-6407 au pavillon Lassonde pendant ces dernières années.

Finalement, merci Marilyne pour ta grande patience et pour tes encouragements pendant les moments les plus difficiles.

Résumé

Cette thèse propose l'étude des codes convolutionnels récurrents doublement orthogonaux (RCDO). Ces codes correcteur d'erreur trouvent leur source dans les travaux développés au cours des dernières années à l'École Polytechnique de Montréal et ont pour objectif de corriger les erreurs se produisant lors du transfert de l'information entre une source et un destinataire. Les codes RCDO représentent une famille de codes convolutionnels récurrents qui offrent des performances d'erreur qui s'approchent des limites prédites par la théorie pour les canaux de communications considérés dans cette thèse.

Outre les excellentes performances d'erreur offertes par les codes RCDO, il s'avère que cette famille de codes correcteurs d'erreur peut être générée par un encodage qui est très simple à réaliser en comparaison avec des techniques de codage qui offrent des performances d'erreur similaires. De plus, le décodage de ces codes s'effectue à l'aide d'un décodeur itératif qui est composé d'une chaîne successive du même décodeur. Le nombre de décodeurs qui se succèdent dans la chaîne représente le nombre d'itérations effectuées lors du décodage. Le fait de répéter la même structure facilite grandement la conception du décodeur itératif, car uniquement la conception d'un seul décodeur doit être prise en compte pour réaliser l'ensemble du décodeur itératif. La simplicité de mise en oeuvre de la technique de codage proposée facilite donc les opérations d'encodage et de décodage et est ainsi adaptée à des sources d'information délivrant des symboles d'information à hauts débits et qui nécessitent de hautes performances d'erreur.

Les objectifs de cette thèse sont multiples. Dans un premier temps, ce travail permet d'établir la correspondance entre les codes Low-Density Parity-Check (LDPC) et les codes RCDO. À partir de cette correspondance, il devient possible de quantifier un seuil de convergence asymptotique associé à une famille de codes RCDO. C'est-à-dire que sous ce seuil, la probabilité d'effectuer une erreur de décodage ne converge pas vers zéro. Nous avons aussi analysé la complexité associée aux opérations liées à l'encodage et au décodage des codes RCDO. Suite à ce travail, il devient donc possible d'imposer des critères de conception matérielle et ainsi générer les codes RCDO qui offrent les meilleurs seuils de convergence théorique.

La question de recherche qui a motivé ce travail est la suivante. Peut-on approcher les limites théoriques associées aux codes correcteurs d'erreur dans des canaux symétriques à l'aide des conditions de double orthogonalité des codes convolutionnels? Pour cette thèse, les canaux symétriques étudiés sont : le canal à bruit additif blanc et gaussien, et le canal binaire à effacement.

Pour répondre à notre question de recherche, nous avons effectué un lien entre le graphe biparti de Tanner des codes convolutionnels doublement orthogonaux et le graphe de Tanner des codes LDPC. À partir de cette correspondance, nous avons utilisé un algorithme appelé, évolution de la densité de probabilité, qui nous a permis d'obtenir par des simulations à l'ordinateur, les seuils de convergence théorique qui correspondent aux codes RCDO qui s'approchent le plus près des limites théoriques. À partir des meilleurs graphes bipartis adaptés aux codes RCDO, nous avons ainsi généré des codes RCDO à l'aide d'heuristiques de recherche. Ces programmes nous ont permis d'obtenir des codes RCDO qui possèdent des seuils de convergence qui s'approchent de la capacité des canaux étudiés.

Les résultats présentés dans cette thèse sont fort encourageants et nous permettent donc de répondre par l'affirmative à notre question de recherche, car les performances d'erreur obtenues par les codes RCDO présentés ne se situent qu'à quelques dixièmes de décibels de la limite de Shannon pour le canal additif à bruit blanc et gaussien. Les résultats obtenus avec les codes RCDO dépassent largement les performances d'erreur des codes convolutionnels non récursifs doublement orthogonaux (CDO) déjà existants, et se comparent aux codes LDPC convolutionnels générés de façon aléatoire que l'on retrouve dans la littérature. Toutefois, à la différence avec ces derniers, la structure des codes RCDO permet de réduire le nombre d'itérations nécessaires lors du décodage tout en facilitant l'encodage. Mentionnons que les codes RCDO sont définis par des conditions de double orthogonalité bien spécifiques tandis que la plupart des codes LDPC convolutionnels sont générés de façon aléatoire. De plus, il s'avère même qu'il existe des codes RCDO qui offrent un gain de codage additionnel d'environ deux décibels par rapport aux codes CDO, et ce, tout en ayant un décodeur qui nécessite moins de calculs à effectuer par symbole d'information décodé. Cependant, le décodeur itératif d'un code RCDO nécessite, dans son ensemble, un nombre supérieur de décodeurs à seuil en comparaison avec un décodeur itératif CDO. De plus, nous présentons aussi un sous-ensemble de codes RCDO qui permet d'atteindre plusieurs taux de codage simultanément à partir d'un code RCDO mère. Ceci représente un

grand avantage pour les systèmes de communications, car en présence d'un canal de communication médiocre il est possible de diminuer facilement le taux de codage du code RCDO mère. À l'inverse en présence d'un canal de communication de bonne qualité il est possible d'augmenter le taux de codage.

Les codes proposés pourraient très bien être utilisés au sein de disques durs électroniques qui nécessitent de bonnes performances d'erreur. Nous pourrions aussi envisager l'utilisation des codes RCDO dans un canal satellitaire permettant ainsi l'échange de l'information tout en minimisant la puissance de transmission nécessaire par les satellites. Ceci aurait pour impact d'allonger la durée de vie des batteries à bord des satellites. Nous pourrions aussi envisager l'utilisation des codes RCDO au sein de terminaux mobiles. En utilisant les codes RCDO, les terminaux mobiles pourraient tirer profit du faible nombre de calculs nécessaires pour encoder et décoder les symboles d'information transmis, et comme pour les satellites, ceci aurait pour impact d'allonger l'autonomie des terminaux mobiles. De plus, les terminaux mobiles sont assujettis à des canaux qui varient continuellement dans le temps. Par conséquent, ces derniers pourraient utiliser des codes RCDO multi taux de codage ce qui permettrait de changer facilement le taux de codage en fonction de la qualité du canal.

Abstract

This thesis presents Recursive Convolutional Doubly-Orthogonal (RCDO) codes. These new error correcting codes represent a class of convolutional Low-Density Parity-Check (LDPC) codes that can be easily decoded iteratively. The doubly-orthogonal conditions of RCDO codes allow the decoder to estimate a symbol with a set of equations that are independent over two successive iterations. This reduces the error propagation throughout the iterative decoding process and therefore improves the error performances. The foundation of this research finds its source in recent works that have been presented in the last decade at École Polytechnique de Montréal.

As presented in this document, the error performances of RCDO codes are near the Shannon capacity for the additive white gaussian noise and the binary erasure channels. In order to achieve these error performances, only a simple multi shift registers recursive convolutional encoder is required at the encoder. Moreover, the iterative decoder is realized only by concatenating, a certain number of times, the same simple threshold decoder. Therefore, the complete decoder is a cascade of the same threshold decoder. It follows that only the design of one simple threshold decoder is needed for constructing the complete iterative decoder. The implementation simplicities of the encoder and of the iterative decoder of RCDO codes is advantageous as compared to the implementation complexity of error correcting techniques that achieve similar error performances.

This thesis has many objectives. First of all, this work presents a bridge between the family of LDPC block codes and the RCDO codes, indeed both families of codes are constructed from their parity-check matrix. From this fact, it becomes possible to identify an asymptotic threshold value that represents the limit above which the error performances of a family of RCDO codes can converge to zero. Moreover, we also present the complexity analysis associated to the encoding and to the decoding of RCDO codes. From this analysis, it becomes now possible to impose material criterions and to search for an ensemble of RCDO codes that meet all the material requirements and have the best theoretical threshold value.

The question that motivates this thesis is the following. Is it possible to approach the theoretical Shannon limits associated to error correcting codes over binary sym-

metric memoryless channels using the doubly-orthogonal conditions imposed on convolutional codes? For this thesis, the symmetric channels are : the Binary Erasure Channel (BEC) and the Additive White Gaussian Noise (AWGN) channel.

To answer that question, we search for a correspondance between the Tanner graph of RCDO codes and the Tanner graph of LDPC block codes. From this correspondance, we use the Density Evolution Algorithm to search, by computer simulations, for RCDO protographs that offer theoretically error performances that converge to zero near the Shannon limits. This search identifies the set of small bipartite graphs, called the protograph, that are used to obtained the complete Tanner graphs associated to an ensemble of RCDO codes. Therefore using an heuristic search, we have found some RCDO codes and the results from our RCDO codes search shows that the threshold value of RCDO codes can approach the Shannon limits for the two considered channels. In particular for the AWGN channel, the error performances of RCDO codes are only at a few tenths of a decibel from the Shannon limits.

It follows that RCDO codes offer error performances that outperform the error performances of the traditional non recursive Convolutional Doubly-Orthogonal (CDO) codes. The additional coding gain offered by RCDO codes, as compared to the CDO codes, is about two decibels. We have also found RCDO codes that require a decoder that need a smaller number of operations per decoded information symbols than one CDO decoder. However, the improvement of the RCDO codes error performances as compared to CDO codes is done at the price of an increase by about 3-fold in the number of iterations that needs to be performed. We also present in this thesis a subclass of RCDO codes, namely the nested RCDO codes. These RCDO codes have the property that the coding rate can be changed easily at the encoder and at the decoder. The multiple coding rates are obtained from one RCDO mother code. Therefore when the channel offers poor conditions, it is possible to decrease the coding rate; at the opposite, when the channel is very good it is also possible to increase the coding rate.

Practically, the proposed RCDO codes can be used in satellite communication systems in order to deliver excellent error performances at low signal-to-noise ratios. One result of this is that the battery life on board the satellites would increase. We can also think to use the nested RCDO codes in wireless communication systems. In such case, a mobile device might change the coding rate of the encoder as a function of the quality of the channel.

Table des matières

Dédicace	iii
Remerciements	iv
Résumé	v
Abstract	viii
Table des matières	x
Liste des tableaux	xiii
Liste des figures	xvi
Liste des annexes	xx
Liste des sigles et des abréviations	xxi
Chapitre 1 INTRODUCTION	1
1.1 Définitions et concepts de base	1
1.2 Éléments de la problématique	2
1.3 Objectifs de recherche	3
1.4 Contributions	3
1.5 Plan de la thèse	4
Chapitre 2 REVUE DE LITTÉRATURE	5
2.1 Codage de canal	5
2.2 Codes Turbo	8
2.3 Codes convolutionnels simplement et doublement orthogonaux	10
2.3.1 Codes convolutionnels simplement orthogonaux	10
2.3.2 Codes convolutionnels doublement orthogonaux	12
2.4 Codes LDPC	15
2.5 Décodage itératif	18

Chapitre 3	Relations entre les codes convolutionnels récurrents RCDO et les codes	
	LDPC	22
3.1	Codes convolutionnels récurrents multiregistres RCDO	23
3.2	Protographes des codes convolutionnels récurrents multiregistres	30
3.3	Graphe biparti de Tanner des codes convolutionnels RCDO	39
3.4	Arbres de calculs associés aux codes convolutionnels RCDO	40
3.5	Conditions d'orthogonalité	43
3.6	Conclusion	45
Chapitre 4	Analyse de la complexité associée au codage et au décodage des codes	
	RCDO	46
4.1	Complexités relatives à l'encodage des codes RCDO	47
	4.1.1 Complexité de calculs liés à l'encodage des codes RCDO	47
	4.1.2 Mémoire requise par un codeur convolutionnel RCDO	49
4.2	Complexités associées au décodage itératif des codes convolutionnels	
	RCDO	50
	4.2.1 Complexité de calculs de l'information extrinsèque	52
	4.2.2 Délai de décodage	56
	4.2.3 Mémoire nécessaire à la réalisation d'une itération du décodeur	
	itératif	56
4.3	Conclusion	57
Chapitre 5	Limites asymptotiques associées aux performances d'erreur des codes	
	RCDO	59
5.1	Seuils de convergence des codes RCDO pour le canal binaire symétrique	
	à effacement	60
	5.1.1 Paires de distributions adaptées aux codes RCDO pour le canal	
	binaire à effacement	63
	5.1.2 Résultats de simulations des codes RCDO pour le canal binaire	
	à effacement	68
5.2	Seuils de convergence des codes RCDO pour le canal binaire symétrique	
	à bruit additif blanc et gaussien	73
	5.2.1 Paires de distributions adaptées aux codes RCDO pour le canal	
	additif à bruit blanc et gaussien	73

5.2.2	Résultats de simulations des codes RCDO pour le canal binaire AWGN non quantifié	86
5.3	Performances d'erreur associées aux codes RCDO réguliers (3,6) dou- blement orthogonaux de faible complexités	91
5.4	Conclusion	96
Chapitre 6	Codes RCDO imbriqués	98
6.1	Introduction aux codes convolutionnels RCDO imbriqués	98
6.2	Protographes et matrices de contrôle des codes RCDO imbriqués . . .	99
6.3	Décodeur itératif mère pour les codes RCDO imbriqués	103
6.4	Performances des codes RCDO imbriqués réguliers $(3, d_\rho)$	105
6.5	Conclusion	109
Chapitre 7	CONCLUSION	110
7.1	Synthèse des travaux	110
7.2	Limitations de la solution proposée	113
7.3	Améliorations futures	114
Références	116
Annexes	120
A.1	Conditions de la double orthogonalité des codes RCDO	120
B.1	Procédure de recherche des valeurs des connexions constituant la ma- trice de contrôle des codes RCDO	127
C.1	Codes RCDO réguliers (3,6)	129
C.2	Code RCDO mère de taux de codage $r = 16/20$	133
C.3	Codes RCDO irréguliers	134

Liste des tableaux

TABLEAU 2.1	Limites de Shannon exprimées en décibel en fonction de différents taux de codage r , pour un canal AWGN lorsqu'une modulation BPSK est utilisée (Lin et Costello, 2004).	8
TABLEAU 5.1	Paire de distributions adaptée aux codes RCDO de taux de codage $r = b/c$ égal à $1/2$ pour le canal BEC, $4 \leq b \leq 10$. On indique aussi pour chaque paire de distributions, la valeur du seuil ϵ^* , ainsi que la différence entre la capacité de Shannon et la valeur du seuil $\Delta\epsilon$	65
TABLEAU 5.2	Paire de distributions adaptée aux codes RCDO de taux de codage $r = b/c$ égal à $1/2$ pour le canal BEC, $11 \leq b \leq 15$. On indique aussi pour chaque paire de distributions, la valeur du seuil ϵ^* , ainsi que la différence entre la capacité de Shannon et la valeur du seuil $\Delta\epsilon$	66
TABLEAU 5.3	Paire de distributions adaptées aux codes RCDO réguliers de taux de codage $r = b/c$ égal à $1/2$ pour le canal BEC. Aussi indiqué pour chaque paire de distributions, la valeur du seuil ϵ^* , le nombre d'arêtes composant le protographe $ E^{\mathcal{P}} $ ainsi que la différence entre la capacité de Shannon et la valeur du seuil $\Delta\epsilon$	68
TABLEAU 5.4	Valeurs des seuils correspondant aux différents codes de la Figure 5.4.	70
TABLEAU 5.5	Seuils correspondant aux ensembles de codes RCDO réguliers de taux de codage $1/2$ possédant des graphes bipartis décrit par la paire de distribution $(x^{d_\lambda-1}, x^{d_\rho-1})$	79
TABLEAU 5.6	Paires de distributions adaptées aux codes convolutionnels récurrents doublement-orthogonaux de taux de codage $r = b/c$ égal à $1/3$, $b \in \{3, 4, 5\}$, offrant de bons seuils de convergence pour le canal AWGN. Pour chaque paire de distribution, le tableau indique la valeur du seuil $(\frac{E_b}{N_0})^*$ en decibel ainsi que le nombre d'arêtes $ E_{\mathcal{P}} $ constituant le protographe.	83

TABLEAU 5.7	Paires de distributions adaptées aux codes convolutionnels récurrents doublement-orthogonaux de taux de codage $r = b/c$ égal à $1/2$, $b \in \{4, 5, 6, 7, 8, 9, 10\}$, offrant de bons seuils de convergence pour le canal AWGN. Pour chaque paire de distribution, le tableau indique la valeur du seuil $(\frac{E_b}{N_0})^*$ en decibel ainsi que le nombre d'arêtes $ E_{\mathcal{P}} $ constituant le protographe.	84
TABLEAU 5.8	Paires de distributions adaptées aux codes convolutionnels récurrents doublement-orthogonaux de taux de codage $r = b/c$ égal à $2/3$, $b \in \{4, 5, 6, 7, 8, 9, 10\}$, offrant de bons seuils de convergence pour le canal AWGN. Pour chaque paire de distribution, le tableau indique la valeur du seuil $(\frac{E_b}{N_0})^*$ en decibel ainsi que le nombre d'arêtes $ E_{\mathcal{P}} $ constituant le protographe.	85
TABLEAU 5.9	Codes convolutionnels LDPC réguliers (3,6) doublement orthogonaux introduisant une faible latence	92
TABLEAU 5.10	Complexité de calculs et mémoires totales nécessaires pour la réalisation des codeurs convolutionnels associés aux codes du Tableau 5.9	92
TABLEAU 5.11	Complexité associée à la réalisation matérielle d'une itération de décodage des codes RCDO du Tableau 5.9.	93
TABLEAU 6.1	Paramètres associés au décodage d'une itération du code RCDO mère (3,15) et du code 1 RCDO du Tableau 5.1.	108
TABLEAU C.1	Matrices de contrôle des codes RCDO réguliers (3, 6) de la Figure 5.11 ayant un taux de codage $r = b/c$ égal à $1/2$, où $b \in \{4, 5, 6\}$	130
TABLEAU C.2	Matrices de contrôle des codes RCDO réguliers (3, 6) de la Figure 5.11 ayant un taux de codage $r = b/c$ égal à $1/2$, où $b \in \{7, 8\}$	131
TABLEAU C.3	Matrices de contrôle des codes RCDO réguliers (3,6) de la Figure 5.11 ayant un taux de codage $r = b/c$ égal à $1/2$, où $b \in \{9, 10\}$	132
TABLEAU C.4	Matrices de contrôle des codes convolutionnels RCDO irréguliers de taux de codage $r = b/c=1/2$, présentés à la Figure 5.4, $b \in \{4, 6, 8, 10\}$	134

TABLEAU C.5 Matrice de contrôle du code convolutionnel RCDO irrégulier.
de taux de codage $r = b/c=1/2$, présentés à la Figure 5.4, $b = 10.135$

Liste des figures

FIGURE 2.1	Schéma bloc représentant un système de communications numériques.	6
FIGURE 2.2	Exemple schématique d'un codeur Turbo de taux de codage $1/3$.	8
FIGURE 2.3	Exemple schématique d'un décodeur pour les codes Turbo. . .	9
FIGURE 2.4	Codeur convolutionnel systématique de taux de codage $r = 1/2$ défini par l'ensemble des connexions $\{0, 1, 3\}$	11
FIGURE 2.5	Exemple d'un codeur convolutionnel non récursif composé de 3 registres en parallèle et d'un taux de codage $3/6$	13
FIGURE 2.6	Exemple d'un codeur convolutionnel récursif composé de 3 registres en parallèle et d'un taux de codage $3/6$	14
FIGURE 2.7	Comparaison des performances d'erreur entre différentes familles de codes convolutionnels de taux de codage $1/2$	15
FIGURE 2.8	Graphe biparti d'un code bloc LDPC régulier	17
FIGURE 2.9	Échange des messages dans un graphe biparti	19
FIGURE 3.1	Exemple d'un codeur convolutionnel récursif composé de 3 registres en parallèle et d'un taux de codage $3/6$	23
FIGURE 3.2	Schéma fonctionnel représentant la formation des équations de contraintes.	25
FIGURE 3.3	Exemple d'un codeur convolutionnel récursif composé de 3 registres en parallèle et d'un taux de codage $3/6$	28
FIGURE 3.4	Protographe général $G_{\mathcal{P}}$ d'un ensemble de code convolutionnels récursifs multi-registres de taux de codage $r = b/c$	32
FIGURE 3.5	Protographe <i>régulier</i> qui représente le code convolutionnel récursif de taux de codage $r = 3/6$ de la Figure 3.3.	34
FIGURE 3.6	Exemple d'un codeur convolutionnel récursif composé de 3 registres en parallèle et d'un taux de codage $r = 3/6$	35
FIGURE 3.7	Protographe <i>irrégulier</i> représentant le code convolutionnel récursif de taux de codage $r = 3/6$ de la Figure 3.6.	36
FIGURE 3.8	Protographe <i>irrégulier</i> correspondant à un ensemble de code convolutionnels non-récursif multi-registres de taux de codage $r = b/c$	38

FIGURE 3.9	Sous-graphe induit par les variables apparaissant dans la n -ième équations de contraintes à l'instant i	39
FIGURE 3.10	a) Sous-graphe induit $\mathcal{T}_{\mathcal{N}_1(v_i^{(m)})}$ b) Sous-graphe induit $\mathcal{T}_{\mathcal{N}_2(v_i^{(m)})}$.	40
FIGURE 3.11	Sous-graphes induits $\mathcal{T}_{\mathcal{N}_1(v_i^{(1)})}$ et $\mathcal{T}_{\mathcal{N}_2(v_i^{(1)})}$ associé à l'exemple 3.4	42
FIGURE 4.1	Procédure d'encodage des symboles de parité au temps i	48
FIGURE 4.2	Schéma bloc représentant le décodeur itératif sur \mathcal{L} itérations des codes RCDO.	51
FIGURE 4.3	Sous-graphe induit $\mathcal{T}_{\mathcal{N}_2(v_t^{(m)})}$ de profondeur 2 au voisinage du noeud variable $v_t^{(m)}$, $t = i - \ell m_s$	52
FIGURE 4.4	Sous-graphe induit $\mathcal{T}_{\mathcal{N}_2(v_t^{(m)})}$ associé à un code RCDO.	53
FIGURE 4.5	Sous-graphe induit $\mathcal{T}_{\mathcal{N}_2(v_t^{(m)})}$ correspondant à l'arbre de décodage du symbole $v_t^{(m)}$ d'un code RCDO.	54
FIGURE 5.1	Canal binaire à effacement.	60
FIGURE 5.2	Évolution de la probabilité d'effacement p_ℓ en fonction du nombre d'itérations ℓ effectuées par le décodeur itératif.	63
FIGURE 5.3	Meilleurs seuils trouvés ϵ^* en fonction du degré maximum alloué aux noeuds variables pour les codes RCDO de taux de codage $r = 1/2$ pour le cas du canal binaire à effacement.	67
FIGURE 5.4	Résultats de simulations pour quatres codes RCDO de taux de codage $1/2$ pour le canal binaire à effacement. Le nombre d'itérations effectués par le décodeur itératif égal à quarante. .	71
FIGURE 5.5	Exemple schématique expliquant le fonctionnement de l'algorithme de l'évolution de la densité de la probabilité pour un graphe ayant une paire de distribution (λ, ρ) . a) Cas pour lequel la probabilité d'erreur tend vers zéro après avoir effectué \mathcal{L} itérations de décodage. b) Cas pour lequel la probabilité d'erreur tend vers une constante après avoir effectué \mathcal{L} itérations de décodage.	77
FIGURE 5.6	Meilleurs seuils trouvés en fonction du degré maximum alloué aux noeuds variables pour les codes RCDO de taux de codage $r = 1/3, 1/2$ et $2/3$ pour le cas du canal AWGN.	80
FIGURE 5.7	Nombre d'additions $ E_{\mathcal{P}} \chi(1)$ requises par itération en fonction du seuil associé aux paires de distributions (λ, ρ) des Tableaux 5.6, 5.7 et 5.8.	82

FIGURE 5.8	Comparaison entre les performances d'erreur de codes convolutionnels récurrents de taux de codage $r = 8/16$ avec celles d'un code bloc LDPC (MacKay, 1999) où $n = 1008$. Pour chaque codes, le graphe biparti possède la paire de distribution (x^2, x^5) .	87
FIGURE 5.9	Performances d'erreur de codes RCDO réguliers (3,6) de taux de codage 15/30 pour plusieurs itérations, $m_s = 149$.	88
FIGURE 5.10	Performances d'erreur de codes convolutionnels RCDO irréguliers de taux de codage $r = 1/4, 1/3, 1/2$ et $2/3$.	90
FIGURE 5.11	Performances d'erreur pour les codes réguliers (3,6) RCDO de taux de codage 1/2 présentés au Tableau 5.9. Aussi reproduit, les performances d'erreur d'un code en bloc régulier (3,6) LDPC de taux de codage 1/2 ainsi que celles d'un code CDO ayant $m_s=1698$ et $r = 1/2$.	95
FIGURE 6.1	Exemple d'un codeur convolutionnel récurrent composé de 3 registres en parallèle et d'un taux de codage 3/6 avec lequel il est possible d'atteindre le taux de codage 2/5.	99
FIGURE 6.2	À gauche, le protographe général $G_{\mathcal{P}}$ d'un ensemble de codes RCDO. À droite, le protographe imbriqué $\tilde{G}_{\mathcal{P}(b-1)}$ qui correspond au sous-graphe induit $G_{\mathcal{P}} \setminus \{v^{(1)}, \dots, v^{(b-1)}\}$.	102
FIGURE 6.3	Protographe $G_{\mathcal{P}}$ d'un code RCDO mère de taux de codage 8/12 offrant le meilleur seuil de convergence asymptotique lorsque $d_{\lambda} = 4$.	103
FIGURE 6.4	Sous-graphes induit $\mathcal{T}_{\mathcal{N}_2(v_i^{(m)})}$ de profondeur deux au voisinage du noeud variables v_i .	104
FIGURE 6.5	Protographe $G_{\mathcal{P}}$ régulier (3, 15) d'un code RCDO mère de taux de codage 16/20.	106
FIGURE 6.6	Performances d'erreur des codes imbriqués RCDO au code mère RCDO régulier (3, 15) qui permet d'atteindre les taux de codage 4/8, 8/12, 12/16, et 16/20 à la suite de 25 itérations, $m_s = 700$. Aussi reproduit, les performances d'un code multi taux de codage CDO avec $m_s = 74451$.	107
FIGURE A.1	a) Sous-graphe induit par les équations (A.1) et (A.2) dans le graphe de Tanner d'un code convolutionnel récurrent simplement orthogonaux. b) Cycle de longueur 4.	122

FIGURE A.2	a) Sous-graphe induit $\mathcal{T}_{\mathcal{N}_4(v_i^{(k)})}$ dans le graphe de Tanner d'un code convolutionnel RCDO. b) Cycle de longueur 6 formé par une chaîne de longueur 2 et d'une chaîne de longueur 4. c) Cycle de longueur 6 formé par 2 chaîne de longueur 3.	123
FIGURE A.3	a) Sous-graphe induit $\mathcal{T}_{\mathcal{N}_4(v_i^{(k)})}$ dans le graphe de Tanner d'un code convolutionnel RCDO. b) Cycle de longueur 6. c) Cycle de longueur 8.	126

Liste des annexes

ANNEXE A	Conditions de la double orthogonalité des codes RCDO	120
ANNEXE B	Méthodes de génération des codes RCDO	127
ANNEXE C	Matrices de contrôle des codes RCDO	129

Liste des sigles et des abréviations

AWGN	Additive White Gaussian Noise
BEC	Binary Erasure Channel
BCJR	Algorithme de décodage de Bahl-Cocke-Jelinek-Raviv
BPSK	Binary Phase Shift Keying
CDO	Convolutional Doubly-Orthogonal codes
LDPC	Low-Density Parity-Check codes
LLR	Log Likelihood Ratio
MLD	Décodeur à maximum de vraisemblance
PEG	Algorithme Progressive Edge Growth
S-CDO	Simplify-Convolutional Doubly-Orthogonal codes
RCDO	Recursive Convolutional Doubly-Orthogonal codes

E_b	Énergie d'un bit (joule)
P	Puissance (Watt)
W	Largeur de bande (Hz)
C	Capacité de Shannon
$\frac{E_b}{N_0}$	Rapport signal sur bruit
ϵ	Probabilité d'effacement d'un canal BEC.
$u_i^{(k)}$	Symbole binaire d'information à la k -ième entrée d'un codeur convolutionnel RCDO à l'instant i
$v_i^{(m)}$	Symbole binaire à la m -ième sortie d'un codeur convolutionnel RCDO à l'instant i
$r = b/c$	Taux de codage d'un code convolutionnel ayant b entrées et c sorties
\mathcal{A}	Ensemble des connexions associé aux codes S-CDO.
$\alpha_{m,n}$	Connexion entre le m -ième symbole à la sortie d'un codeur RCDO et le n -ième registre à décalage composant le codeur.
m_s	Mémoire du plus grand registre à décalage composant un codeur convolutionnel
\mathbf{H}^T	Matrice de contrôle binaire associée à un code LDPC
\mathbf{v}	Vecteur à composantes binaires représentant un mot de code.
G	Graphe biparti de Tanner
V_c	Ensemble des noeuds de contraintes composant le graphe biparti de Tanner.
V_v	Ensemble des noeuds variables composant le graphe biparti de Tanner.
E	Ensemble des arêtes composants le graphe biparti de Tanner.
$d(c)$	Degré associé au noeud c dans un graphe
$\mathcal{N}(c)$	Ensemble des noeuds qui possèdent une arête en commun avec le noeud c dans un graphe.
d_λ	Degré maximum associé aux noeuds variables composant le graphe biparti de Tanner.

d_ρ	Degré maximum associé aux noeuds de contraintes composant le graphe biparti de Tanner.
\bar{d}_λ	Degré moyen associé aux noeuds variables composant le graphe biparti de Tanner.
\bar{d}_ρ	Degré moyen associé aux noeuds de contraintes composant le graphe biparti de Tanner.
$\lambda(x)$	Fonction d'énumération représentant les fractions des arêtes incidentes aux noeuds variables d'un certain degré dans le protographe.
$\rho(x)$	Fonction d'énumération représentant les fractions des arêtes incidentes aux noeuds de contrainte d'un certain degré dans le protographe.
(λ, ρ)	Paire de distributions associée aux fonctions d'énumération $\lambda(x)$, $\rho(x)$.
$\mathbf{H}^T(D)$	Matrice de contrôle associée à un code RCDO.
$G_{\mathcal{P}}$	Graphe biparti appelé Protographe qui est associé à un code RCDO.
$V_c^{\mathcal{P}}$	Ensemble des noeuds de contraintes composant un protographe.
$V_v^{\mathcal{P}}$	Ensemble des noeuds variables composant un protographe.
$E_{\mathcal{P}}$	Ensemble des arêtes composant un protographe.
$m_v^{(\ell)}$	Message à la ℓ -ième itération qui correspond au noeud v .
$m_{vc}^{(\ell)}$	Message échangé à la ℓ -ième itération entre le noeud v et le noeud c du graphe de Tanner.
\mathcal{R}	Opérateur permettant lors du décodage d'effectuer la mise à jour des messages entre les noeuds de V_c et les noeuds de V_v dans le graphe de Tanner.
\mathcal{T}_c	Sous-graphe induit du graphe de Tanner représentant un arbre dont la racine est le noeud c .
N_Σ^c	Nombre d'additionneurs modulo-2 nécessaires à la réalisation d'un codeur RCDO.
N_Σ^d	Nombre d'additionneurs de nombre réels nécessaires à la réalisation d'une itération.
$N_{\mathcal{R}}$	Nombre d'opérateurs \mathcal{R} nécessaires à la réalisation d'une itération de décodage.

Chapitre 1

INTRODUCTION

La recherche proposée dans cette thèse étudie un nouveau type de codes correcteurs d'erreur permettant de diminuer l'impact des perturbations introduites (bruit) par un canal de transmission au sein d'un système de communications numériques. L'idée est simple et consiste à ajouter des symboles de parité aux symboles d'information transmis. En d'autres termes, l'émetteur transmet plus de bits qu'il n'en faut pour représenter la source numérique : les bits d'information et les symboles de parité. Les symboles de parité sont obtenus à la sortie d'un codeur de canal qui utilise les symboles d'information ainsi que certaines règles pour générer les symboles de parité. Une fois les données codées, elles sont transmises vers un destinataire. À la réception, les symboles de parité sont utilisés par un décodeur pour estimer les symboles d'information qui sont en fait les symboles utiles. Le but de notre recherche est d'établir une nouvelle méthode de codage qui permet de s'approcher pratiquement des limites annoncées par la théorie.

1.1 Définitions et concepts de base

Les codes correcteur d'erreur envisagés dans ce projet représentent une fusion entre deux familles de codes correcteur d'erreur déjà existantes : les codes Low Density Parity Check Codes (LDPC) (Gallager, 1962), (MacKay, 1999) et les codes convolutionnels beaucoup plus anciens. La combinaison proposée simplifie grandement l'encodage des codes LDPC en bloc à cause de la structure convolutionnelle des codes. Au récepteur, le décodeur estime les symboles d'information à l'aide d'un algorithme itératif qui, par un mécanisme d'inférence probabiliste, prend une décision dure (1 ou 0) sur le symbole d'information transmis par la source. Ce processus de décodage itératif est basé sur l'échange de messages dans le graphe biparti des nouveaux codes proposés. Toutefois, pour effectuer des estimations fiables lors du décodage, nous devons nous assurer que, d'une itération à l'autre, les estimations

effectuées soient indépendantes les unes des autres. Pour obtenir cette indépendance, les codes convolutionnels récurrents multiregistres utilisent le principe de la double orthogonalité développé dans (Cardinal, 2001) et (Cardinal *et al.*, 2003).

La double orthogonalité des codes convolutionnels est un concept qui trouve sa source dans les travaux de Massey (Massey, 1963). Dans les années 1960, James Massey a eu l'idée d'utiliser les codes convolutionnels et d'imposer des conditions algébriques à la position des connexions du codeur convolutionnel. Les conditions imposées par Massey permettent l'utilisation d'un simple décodeur à seuil pour décoder les codes convolutionnels. La méthode de décodage proposée par Massey est sous-optimale. En faisant évoluer le principe de Massey et en utilisant des conditions plus restrictives sur la position des connexions, qui composent le codeur convolutionnel, permet l'utilisation d'un décodeur itératif qui est composé d'une chaîne de décodeurs identiques à celui proposé par Massey. De cette manière, en utilisant les codes convolutionnels doublement orthogonaux (CDO), nous pouvons augmenter le gain de codage par rapport aux codes convolutionnels de Massey.

L'analyse des structures des codes convolutionnels récurrents doublement orthogonaux (RCDO) n'a jamais été entreprise jusqu'à présent et nous verrons que les résultats obtenus montrent que les codes présentés offrent des performances d'erreur qui se comparent favorablement aux techniques de correction d'erreur les plus performantes à l'heure actuelle.

1.2 Éléments de la problématique

Cette thèse a pour objectif de répondre à la problématique liée à la construction des codes convolutionnels RCDO et à leur décodage. C'est-à-dire que nous désirons établir une technique de correction des erreurs qui soit simple à réaliser tout en offrant des performances d'erreur proches des limites théoriques. Pour ce faire, nous devons établir un moyen d'évaluer les performances d'erreur à faible rapport signal sur bruit dans le but de construire des codes RCDO performants et ainsi les comparer aux meilleures techniques actuelles. Pour établir les limites théoriques associées aux codes RCDO, nous devons établir *a priori* un modèle qui permet de trouver ces limites.

Une fois ce modèle établi, nous devons évaluer les performances d'erreur asymptotiques à faibles rapports signal-sur-bruit des nouveaux codes RCDO. L'évaluation des performances d'erreur asymptotiques s'effectue à l'aide d'un algorithme que nous

pouvons retrouver dans la littérature et qui nécessite des simulations par ordinateurs. À partir de cet algorithme, nous devons effectuer la recherche des codes RCDO qui offrent des performances d'erreur aussi près que possible de la capacité de Shannon. Par conséquent, nous devons aussi effectuer la recherche des codes RCDO. Pour ce faire, nous avons établi une heuristique de recherche qui permet de générer des codes RCDO.

1.3 Objectifs de recherche

Les objectifs de la recherche proposée s'articulent autour de deux axes. Le premier axe définit essentiellement les limites théoriques tandis que le second axe vise plutôt l'élaboration de simulateurs qui permettent la recherche des nouveaux codes RCDO. Dans un premier temps, nous désirons établir les limites asymptotiques à faibles rapports signal sur bruit qui peuvent être associées aux codes convolutionnels RCDO. Outre les limites asymptotiques, nous désirons aussi établir la complexité qui est rattachée à l'encodage et au décodage itératif de ces codes ce qui nous permet de comparer la méthode proposée à d'autres techniques modernes de correction des erreurs. Les analyses effectuées dans cette thèse ont pour objectifs d'établir un canevas qui permet de générer les meilleurs codes RCDO qui satisfont les critères matériels imposés.

1.4 Contributions

Les contributions apportées par ce travail de recherche sont les suivantes :

1. Établissement d'un lien entre les codes en blocs LDPC et les codes convolutionnels RCDO.
2. Analyse de la complexité liée à l'encodage et au décodage itératif des codes convolutionnels RCDO.
3. Recherche des protographes associés aux codes RCDO qui permettent d'approcher la capacité des canaux binaires symétriques et sans mémoire BEC et AWGN.

4. Conception de programmes permettant la prospection des protographes associés aux codes RCDO.
5. Conception de programmes permettant la recherche des codes RCDO possédant un certain protographe.
6. Élaboration de la définition des codes imbriqués RCDO offrant plusieurs taux de codage.

Toutes les simulations ont été effectuées à l'aide d'ordinateurs munis de processeurs AMD PhenomTM II \times 4 955 cadencés à 3.2 GHz et possédant 4 Gb de RAM sous l'environnement Windows7[®]. Les logiciels de programmation Matlab[®] et MicrosoftVisualStudio[®] ont servi à la conception des simulateurs utilisés.

1.5 Plan de la thèse

Cette thèse est divisée en sept chapitres. Le second chapitre présente une revue de littérature dans laquelle nous exposons certaines techniques de codage modernes qui représentent les fondements sur lesquels repose ce document. Au troisième chapitre, nous définissons les codes convolutionnels récurrents multiregistres doublement orthogonaux (RCDO). Au quatrième chapitre, nous effectuons l'évaluation de la complexité de calculs associée à l'encodage ainsi qu'au décodage des codes RCDO. Au chapitre cinq, nous précisons les meilleures structures qui offrent théoriquement les meilleures performances asymptotiques à faible rapport signal-sur-bruit associées aux codes RCDO. Au chapitre 6, nous identifions les codes RCDO mères qui permettent de générer des codes RCDO imbriqués de différents taux de codage. Au dernier chapitre, nous concluons cette recherche en présentant les limitations de ce travail ainsi que les ouvertures qui nous semblent les plus prometteuses pour le futur.

Chapitre 2

REVUE DE LITTÉRATURE

Dans ce chapitre, nous effectuons une revue de la littérature qui survole les techniques modernes de correction des erreurs qui ont inspiré notre travail. Nous discutons plus précisément du théorème du codage de canal de Shannon, des codes Turbo ainsi que des codes convolutionnels orthogonaux et doublement orthogonaux. Finalement, nous présentons les codes linéaires en blocs LDPC ainsi que l'algorithme itératif à passage de messages.

2.1 Codage de canal

En 1948, Claude Shannon a présenté le théorème du codage de canal dans un travail pionnier (Shannon, 1948) qui a défini un énorme domaine de recherche. Non seulement dans ce travail nous pouvons y retrouver les définitions associées aux différentes mesures de l'information, mais nous pouvons aussi y retrouver les limites les plus importantes qui régissent les systèmes de communications numériques que nous utilisons couramment.

À la Figure 2.1, nous présentons le schéma en bloc qui est utilisé pour modéliser le système de communication que nous avons considéré dans cette thèse. En se référant à cette figure, une source émet à chaque instant i des symboles d'information binaires u_i qui correspondent au message à transmettre vers le destinataire. Shannon a démontré qu'en ajoutant un codeur de canal de taux de codage $r = k/n$, $k < n$, il devient possible de contrôler les erreurs qui peuvent être provoquées par le canal de transmission. Un codeur de canal de taux de codage $r = k/n$ utilise k symboles d'information binaires provenant de la source pour générer à sa sortie un groupe de n symboles binaires codés. Les n symboles codés forment un mot de code $\mathbf{v} = (v_1, \dots, v_n)$ qui est obtenu en effectuant une combinaison linéaire de k vecteurs de dimension n . Les symboles codés à la sortie du codeur de canal sont transmis vers un modulateur qui sert à convertir les valeurs binaires logiques en signaux analogiques $s(t)$ pouvant être trans-

mis dans le canal de transmission. Dans cette thèse, nous considérons uniquement un modulateur BPSK, qui effectue une modulation antipodale. Les symboles $s_i(t)$ sont obtenus en utilisant la règle d'assignation suivante $s_i(t) = x_i f(t)$ où $x_i = (1 - 2v_i)$ où $f(t)$ représente une fonction d'énergie E_s qui est définie sur le support $0 \leq t \leq T$. Ces symboles sont ensuite transmis vers un canal binaire symétrique sans mémoire à bruit additif blanc et gaussien (AWGN). Le signal reçu au récepteur $r_i(t) = s_i(t) + n_i(t)$ représente une version du signal $s_i(t)$ qui est corrompue par un bruit additif $n_i(t)$ qui correspond à : un processus aléatoire gaussien de moyenne nulle et de densité de puissance spectrale bilatérale égale à $N_0/2$ (Watt/Hz).

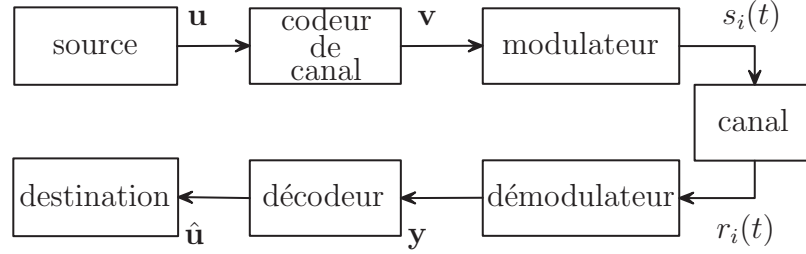


FIGURE 2.1 Schéma bloc représentant un système de communications numériques.

Le démodulateur produit à la sortie d'un filtre adapté normalisé une valeur réelle y_i qui correspond :

$$y_i = \sqrt{E_s}(1 - 2v_i) + n_i \quad (2.1)$$

où n_i représente une variable aléatoire gaussienne de moyenne nulle et de variance $N_0/2$. Ces valeurs sont alors utilisées par le décodeur dans le but de produire une estimation \hat{u}_i des symboles d'information transmis u_i .

L'une des limites établie par Shannon est celle liée à la capacité théorique de transmission dans un canal additif à bruit blanc et gaussien de moyenne nulle et de densité spectrale bilatérale égale à $\frac{N_0}{2}$. Shannon a montré que pour ce type de canal la capacité théorique en bits/s est égale à :

$$C = W \log_2 \left(1 + \frac{P}{WN_0} \right) \quad (2.2)$$

où P (Watt) désigne la puissance des signaux, et W (Hz) représente la largeur de bande du canal de communication. À partir de l'équation 2.2, Shannon a établi le

théorème du codage de canal. Ce dernier à la signification suivante : si le débit d'information à la sortie de la source est inférieur à la capacité du canal C alors il est théoriquement possible, en utilisant un code correcteur d'erreur approprié, d'effectuer une transmission de l'information sans erreur. Par contre, si le débit de la source est supérieur à la capacité du canal alors quelque soit le code correcteur d'erreur utilisé, il ne sera pas possible d'obtenir une transmission de l'information sans erreur.

Cela dit, le théorème du codage de canal de Shannon est souvent référé comme étant une preuve non constructive au sens où il ne nous indique en rien la manière dont nous devons nous y prendre pour atteindre les performances prédites par la théorie. Le théorème de Shannon nous indique qu'il existe des codes correcteurs d'erreur ainsi qu'un algorithme de décodage qui permet de faire tendre la probabilité d'erreur vers zéro. Suite à ces résultats, un très grand nombre de chercheurs ont entrepris la recherche de techniques de codage permettant de s'approcher des limites proposées dans (Shannon, 1948). À partir de l'équation (2.2), il est possible d'établir le rapport signal-sur-bruit $\frac{E_b}{N_0}$ minimum à partir duquel il est possible de transmettre sans erreur dans un canal AWGN. Lorsque nous considérons une largeur de bande infinie, cette valeur $\left(\frac{E_b}{N_0}\right)_{\min}$ est appelée la limite de Shannon et correspond à -1.6 dB (Shannon, 1948). L'interprétation de cette limite est qu'il n'existe aucun code correcteur d'erreur qui permet de faire tendre la probabilité d'erreur vers zéro lorsque le rapport signal sur bruit $\frac{E_b}{N_0}$ est inférieur à -1.6 dB. Nous pouvons aussi montrer que le rapport $\frac{E_b}{N_0}$ minimum varie en fonction du taux de codage r des codes correcteur d'erreur utilisés. Le Tableau 1.1 indique la valeur du rapport $\frac{E_b}{N_0}$ minimum lorsqu'on envisage l'utilisation de codes correcteur d'erreur de taux de codage $r = 1/10, 1/4, 1/3, 1/2, 2/3, 3/4$ et $4/5$ pour un canal binaire symétrique AWGN à sortie non quantifiée. En se référant à ce tableau, il est théoriquement possible de développer une technique correctrice d'erreur de taux de codage $r = 1/2$ qui permet de faire tendre la probabilité d'erreur vers 0 pour des valeurs de $\frac{E_b}{N_0}$ supérieures à 0.188 dB. Nous utiliserons donc dans cette thèse, les limites de Shannon présentées au Tableau 1.1 comme valeurs de référence pour comparer les performances d'erreur des codes correcteurs d'erreur présentés dans cette thèse.

Il a fallut attendre près de cinquante ans avant de voir émerger le premier mécanisme de contrôle des erreurs qui a permis de s'approcher à quelques dixièmes de décibels de la capacité de Shannon. Ces codes correcteurs d'erreur, appelés les codes Turbo, furent présentés originalement dans (Berrou *et al.*, 1993).

TABLEAU 2.1 Limites de Shannon exprimées en décibel en fonction de différents taux de codage r , pour un canal AWGN lorsqu'une modulation BPSK est utilisée (Lin et Costello, 2004).

Taux de codage	Rapport $\frac{E_b}{N_0}$ (en dB)
1/10	-1.548
1/4	-0.793
1/3	-0.507
1/2	0.188
2/3	1.021
3/4	1.628
4/5	2.045

2.2 Codes Turbo

Dans cette section, nous résumons qualitativement les principes associés à l'encodage et au décodage itératif des codes Turbo. En nous référant à la Figure 2.2, nous pouvons décrire le principe du codage Turbo qui consiste à encoder une séquence binaire à l'aide de deux codeurs convolutionnels en parallèle. Le premier codeur convolutionnel encode la séquence binaire provenant directement de la source, tandis que le second codeur convolutionnel encode la séquence binaire à la sortie d'un entrelaceur, représenté par le symbole π sur la figure. Le rôle de l'entrelaceur est de permuter les bits de la séquence binaire qui provient de la source. Le but de cette opération est de faire en sorte que le second codeur convolutionnel encode une séquence de symboles qui apparaît différente de celle encodée par le premier codeur convolutionnel.

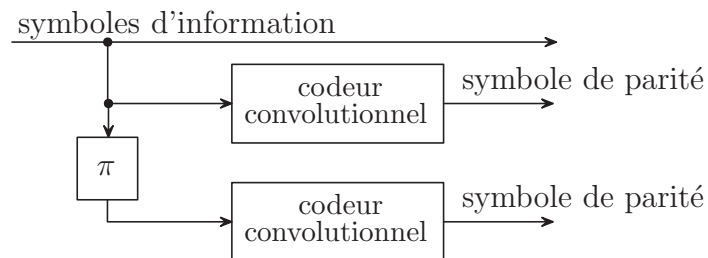


FIGURE 2.2 Exemple schématique d'un codeur Turbo de taux de codage 1/3.

À la Figure 2.3, nous présentons un schéma fonctionnel qui explique le décodage des codes Turbo. Comme nous pouvons le constater sur cette figure, le décodeur est constitué de deux décodeurs qui s'échangent de l'information. Le premier décodeur calcule des valeurs de fiabilité à partir de la séquence d'information provenant du canal ainsi qu'à partir de la séquence de parité provenant du canal qui est générée par le premier codeur convolutionnel. Le second décodeur calcule lui aussi certaines valeurs de fiabilité sur les symboles d'information transmis, mais à partir de la séquence d'information provenant du canal ainsi que des symboles de parité provenant du second codeur convolutionnel. Une fois que les deux décodeurs ont calculé des valeurs de fiabilité sur l'ensemble des symboles d'information, ils s'échangent l'information extrinsèque qui correspond à chacun des symboles. Le nombre de fois où l'on échange de l'information extrinsèque entre les deux décodeurs correspond au nombre d'itérations effectuées par le décodeur.

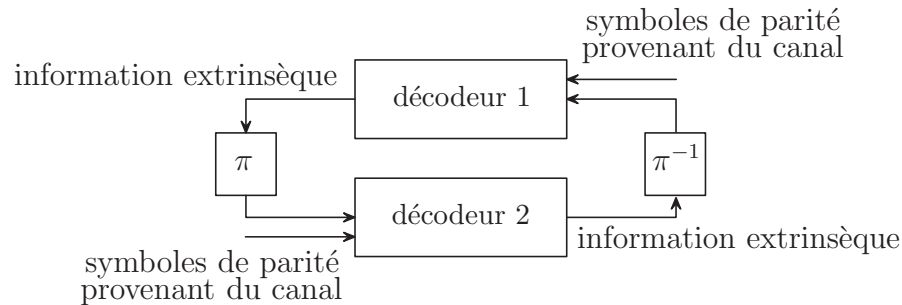


FIGURE 2.3 Exemple schématique d'un décodeur pour les codes Turbo.

L'algorithme utilisé dans (Berrou *et al.*, 1993) pour calculer les valeurs de fiabilité échangées lors du décodage des codes Turbo est l'algorithme BCJR (Bahl *et al.*, 1976) qui effectue une estimation qui maximise la probabilité *a posteriori* des symboles transmis. Le succès des codes Turbo repose essentiellement sur le type et la taille de l'entrelaceur qui est utilisé (Lin et Costello, 2004). Généralement, plus la taille de l'entrelaceur est élevée plus les performances d'erreur de ces codes convolutionnels s'approchent de la limite de Shannon. En contrepartie, une taille élevée de l'entrelaceur nécessite une accumulation d'un très grand nombre de symboles et par conséquent génère une certaine latence lors de l'encodage et du décodage de la séquence d'information. Or, ceci peut s'avérer néfaste dans certains systèmes de

communications sensibles au délai.

À la suite de la découverte des codes Turbo, un très grand nombre de chercheurs sont venus dynamiser la recherche dans ce domaine d'expertise. Plusieurs questions d'intérêts théoriques ont émergé suite à cette découverte et des résultats théoriques importants ont été établis. De plus, plusieurs travaux ont été redécouverts. Par exemple, dans (Wiberg, 1996) on propose de voir les codes linéaires sous une forme de graphe biparti appelé le graphe de Tanner (Tanner, 1981). D'ailleurs, c'est en considérant un graphe biparti pour représenter un code en blocs que les auteurs de (MacKay, 1999) et (Luby *et al.*, 1997) ont redécouvert les codes en blocs LDPC qui avait été proposés par Gallager dans sa thèse de doctorat (Gallager, 1962). Au même moment, l'idée de la double orthogonalité des codes convolutionnels naquit et fut présentée dans le but d'éliminer l'entrelaceur qui apparaît dans la technique de codage Turbo (Cardinal *et al.*, 1998), (Cardinal, 2001).

2.3 Codes convolutionnels simplement et doublement orthogonaux

2.3.1 Codes convolutionnels simplement orthogonaux

Dans cette section, nous exposons la définition des codes convolutionnels simplement orthogonaux (CSO) présentée dans (Massey, 1963), et nous présentons les conditions de la double orthogonalité des codes convolutionnels (Cardinal, 2001). À la Figure 2.4, nous présentons un codeur convolutionnel systématique de taux de codage $1/2$. Ce codeur convolutionnel accepte, à chaque instant i , un symbole binaire $u_i^{(1)}$ et génère deux symboles binaires $v_i^{(1)}$ et $v_i^{(2)}$. Le codeur convolutionnel est composé d'additionneurs modulo 2 et d'un registre à décalage ayant m_s éléments de délai. La propriété des codes CSO vient du fait que le symbole $v_i^{(1)}$ est connecté à certains éléments de délai qui constituent le registre à décalage. Nous pouvons caractériser complètement un code CSO à partir de l'ensemble des connexions $\mathcal{A} = \{\alpha_1, \dots, \alpha_{|\mathcal{A}|}\}$ où la valeur α_k représente la k -ième connexion entre le symbole $v_i^{(1)}$ et le α_k -ième élément de délai du registre à décalage, $\alpha_k \in \mathbb{N}$, $k = \{1, \dots, |\mathcal{A}|\}$ et $\alpha_{|\mathcal{A}|} = m_s$.

Par exemple, le code CSO de la Figure 2.4 est défini par l'ensemble $\mathcal{A} = \{0, 1, 3\}$, car il y a trois connexions entre le symbole $v_i^{(1)}$ et le registre à décalage. La première connexion α_1 prend la valeur zéro, car elle se situe avant le premier élément de délai.

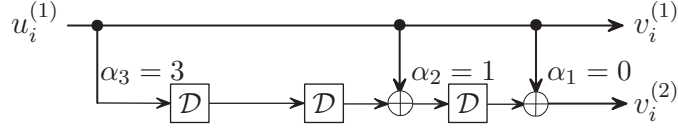


FIGURE 2.4 Codeur convolutionnel systématique de taux de codage $r = 1/2$ défini par l'ensemble des connexions $\{0, 1, 3\}$.

La seconde connexion α_2 est égale à un, car elle est située après le premier élément de délai, et la dernière connexion α_3 est égale à trois car elle est connectée à la suite du troisième élément de délai. Avec l'ensemble \mathcal{A} , nous pouvons exprimer la valeur binaire $v_i^{(2)}$ à la sortie du codeur convolutionnel. Cette valeur représente une combinaison linéaire des symboles binaires précédemment transmis et peut s'écrire :

$$v_i^{(2)} = \sum_{k=1}^{|\mathcal{A}|} v_{i-\alpha_k}^{(1)} \quad (2.3)$$

où la somme dans l'équation (2.3) représente une somme modulo 2. Massey a montré que nous pouvons utiliser un simple décodeur à seuil si la définition suivante est respectée par l'ensemble de connexions \mathcal{A} qui définit le codeur convolutionnel.

Définition 2.1 : Un code convolutionnel systématique est *simplement orthogonal* si la position des connexions du codeur est telle que :

Les différences $(\alpha_k - \alpha_l)$ sont distinctes, $k \neq l$, $\alpha_k, \alpha_l \in \mathbb{N}$, l et $k \in \{1, 2, \dots, |\mathcal{A}|\}$.

Cette définition permet en fait d'effectuer le décodage des symboles à l'aide d'un ensemble d'équations indépendantes les unes des autres. C'est-à-dire qu'il est possible d'inférer la valeur d'un symbole d'information à partir de $|\mathcal{A}|$ équations qui font intervenir des symboles différents les uns des autres. Les résultats théoriques et pratiques nous montrent que les performances d'erreur asymptotiques, à haut rapport signal sur bruit, varient en fonction du nombre de connexions qui constituent l'ensemble \mathcal{A} . C'est-à-dire que pour de grandes valeurs de $\frac{E_b}{N_0}$, plus le nombre de connexions dans l'ensemble \mathcal{A} est grand plus les performances d'erreur obtenues avec les codes CSO s'améliorent. L'algorithme de décodage proposé par Massey est très simple à réaliser

à partir d'un simple décodeur à seuil. Ce type de décodeur est sous-optimum par rapport à l'algorithme de décodage de Viterbi qui est généralement utilisé pour effectuer le décodage par séquence des codes convolutionnels (Lin et Costello, 2004).

2.3.2 Codes convolutionnels doublement orthogonaux

Les conditions de la double orthogonalité présentées dans (Cardinal *et al.*, 1998) ont pris naissance lorsque les auteurs ont décidé d'utiliser une cascade du décodeur à seuil présenté par Massey. Cette opération a pour objectif de conserver la structure itérative du décodage des codes Turbo, mais d'éliminer les entrelaceurs que l'on retrouve au sein de la technique de codage Turbo (Cardinal *et al.*, 1998). Pour les codeurs convolutionnels systématiques ne possédant qu'un seul registre à décalage, les conditions de la double orthogonalité ne peuvent pas être complètement remplies. Ces codes sont définis comme étant des codes convolutionnels doublement orthogonaux simplifiés (S-CDO) (Roy *et al.*, 2007).

Définition 2.2 : Un code convolutionnel systématique de taux de codage $1/2$ est doublement orthogonal et simplifié (S-CDO) si et seulement si l'ensemble des connexions $\mathcal{A} = \{\alpha_k, k = 1, \dots, |\mathcal{A}|\}$ répond aux conditions suivantes :

1. Les différences $(\alpha_k - \alpha_l)$ doivent être distinctes,
2. Les différences de différences $(\alpha_k - \alpha_l) - (\alpha_m - \alpha_n)$ peuvent être égales,
3. Les différences générées par la condition 1 doivent être distinctes des différences de différences générées par la condition 2.

où $k \neq l, m \neq n, k \neq m, l \neq n$ et $\alpha_k \in \mathbb{N}$.

À la Figure 2.7 nous avons indiqué les performances d'erreur d'un code S-CDO de taux de codage $1/2$ après 8 itérations (Roy *et al.*, 2007). L'ensemble des connexions \mathcal{A} de ce code comporte 10 éléments et le registre à décalage est composé de 340 éléments de délai. Comme nous pouvons le remarquer, ce type de code permet d'améliorer grandement les performances d'erreur par rapport au scénario sans l'utilisation d'un code correcteur d'erreur. Pour une probabilité d'erreur par bit de 10^{-4} , l'utilisation du code correcteur d'erreur S-CDO permet un gain de codage d'environ 6 dB par rapport

au scénario sans codage. À titre comparatif, nous avons reproduit sur la figure les performances d'erreur du code Turbo, de taux de codage $1/2$, présenté dans (Berrou *et al.*, 1993). Ces performances d'erreur du code Turbo ont été obtenues après avoir effectué 18 itérations de décodage et avec un entrelaceur qui effectue la permutation d'un bloc de 65536 bits. Comme nous pouvons le constater ce code permet de faire tendre la probabilité d'erreur par bit vers zéro lorsque le rapport signal-sur-bruit est supérieur à 0.6 dB. Cette valeur est très près, en se référant au tableau 1.1, de la limite de Shannon pour ce taux de codage. Généralement, les performances d'erreur obtenues avec les codes S-CDO se retrouvent assez éloignées des performances d'erreur obtenues avec les codes Turbo. Beaucoup d'efforts ont été mis sur la recherche de codes S-CDO. Par exemple, les travaux de recherche qui se retrouvent dans (Baechler *et al.*, 2000) et (Baechler, 2000) présentent les premiers codes S-CDO. Tandis que les travaux présentés dans (Roy *et al.*, 2007) et (Cardinal *et al.*, 2009) généralisent la définition des codes S-CDO. De façon générale, tous les codes S-CDO utilisant un décodeur itératif à seuil comme celui présenté dans (Cardinal, 2001) nécessitent environ 8 itérations de décodage avant d'atteindre un point de saturation où l'ajout d'itérations ne permet plus d'améliorer les performances d'erreur.

Contrairement aux codes S-CDO, qui ont été largement étudiés ces dernières années, les codes convolutionnels systématiques réellement doublement orthogonaux (CDO) présentés dans (Cardinal, 2001) ont été largement moins étudiés. Les codes CDO de taux de codage $r = b/c$ sont générés à partir de codeurs convolutionnels qui possèdent $(c - b)$ registres à décalage en parallèle comme celui présenté à la Figure 2.5. À chaque instant i , le codeur accepte b symboles binaires $u_i^{(k)}, k \in \{1, \dots, b\}$ et génère c symboles binaires à la sortie $v_i^{(m)}, m \in \{1, \dots, c\}$.

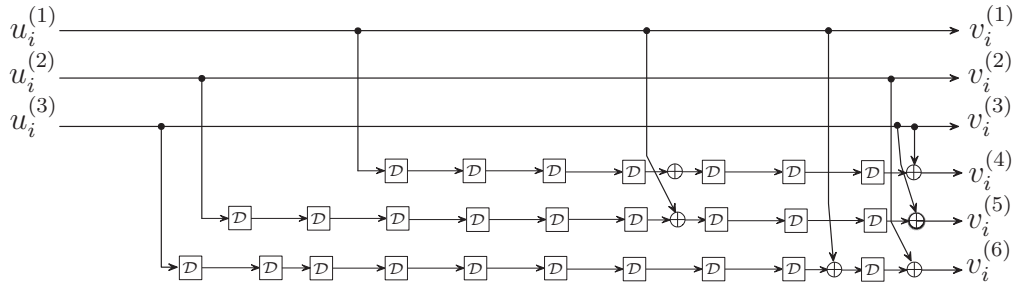


FIGURE 2.5 Exemple d'un codeur convolutionnel non récursif composé de 3 registres en parallèle et d'un taux de codage $3/6$.

Notons que chaque symbole d'information à l'entrée du codeur ne peut être injecté plus d'une fois dans le même registre. Cette condition permet de préserver la double orthogonalité des codes convolutionnels (Cardinal, 2001). Les codes CDO permettent d'estimer un symbole à l'aide d'équations complètement indépendantes sur deux itérations. Toutefois, les codes CDO offrent des performances d'erreur très similaires à celles que nous pouvons obtenir avec les codes S-CDO (He *et al.*, 2009).

En fait, c'est en ajoutant des boucles de rétroactions au sein du codeur convolutionnels multiregistres que les meilleurs codes doublement orthogonaux ont été observés. C'est dans (Cardinal, 2001) que l'on définit pour la première fois les codes récursifs doublement orthogonaux et récemment ces résultats furent publiés dans (Cardinal *et al.*, 2008). À la Figure 2.6, nous présentons un exemple de codeur convolutionnel récursif multiregistres. Le fonctionnement du codeur est identique à celui du codeur CDO. Toutefois, contrairement au codeur CDO de la Figure 2.5, nous réinjectons les symboles de parité dans les $(c - b)$ registres à décalage en parallèle qui constitue le codeur RCDO.

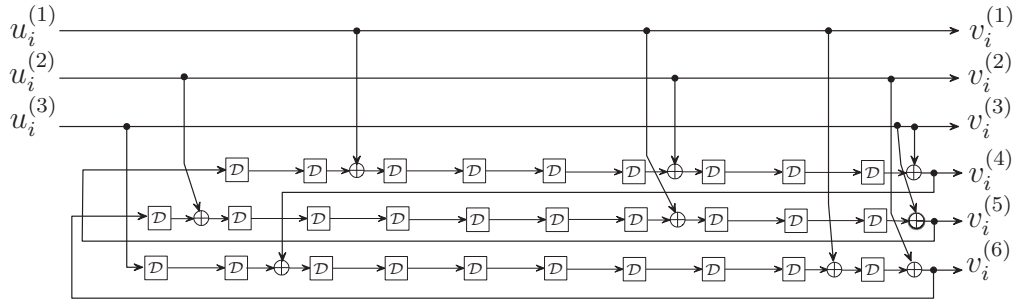


FIGURE 2.6 Exemple d'un codeur convolutionnel récursif composé de 3 registres en parallèle et d'un taux de codage 3/6.

À la Figure 2.7 nous présentons les résultats de simulations des deux codes RCDO présentés dans (Cardinal, 2001). Comme nous pouvons le remarquer, les codes RCDO A et B offrent des performances d'erreur bien meilleures que celles obtenues par le code S-CDO. Particulièrement, le code RCDO-B s'approche à environ 1dB de la limite de Shannon. L'auteur de ces codes RCDO a généré le code B à partir du code A, en conservant exactement le même codeur que le code A, mais en éliminant de façon aléatoire des connexions composant le codeur convolutionnel multiregistres. Plusieurs tentatives infructueuses ont dû être effectuées pour générer le code RCDO-B, et c'est en quelque sorte par hasard que de tels résultats ont été obtenus. Un des objectifs de

cette thèse est précisément de répondre à cette problématique. Soit celle d'identifier certaines structures adaptées aux codes RCDO qui permette d'approcher la limite de Shannon. Pour identifier ces structures, nous effectuons au chapitre suivant, la liaison entre les codes RCDO et les codes LDPC en bloc. Mais avant de faire cette liaison, nous définissons d'abord certaines caractéristiques associées aux codes LDPC en blocs.

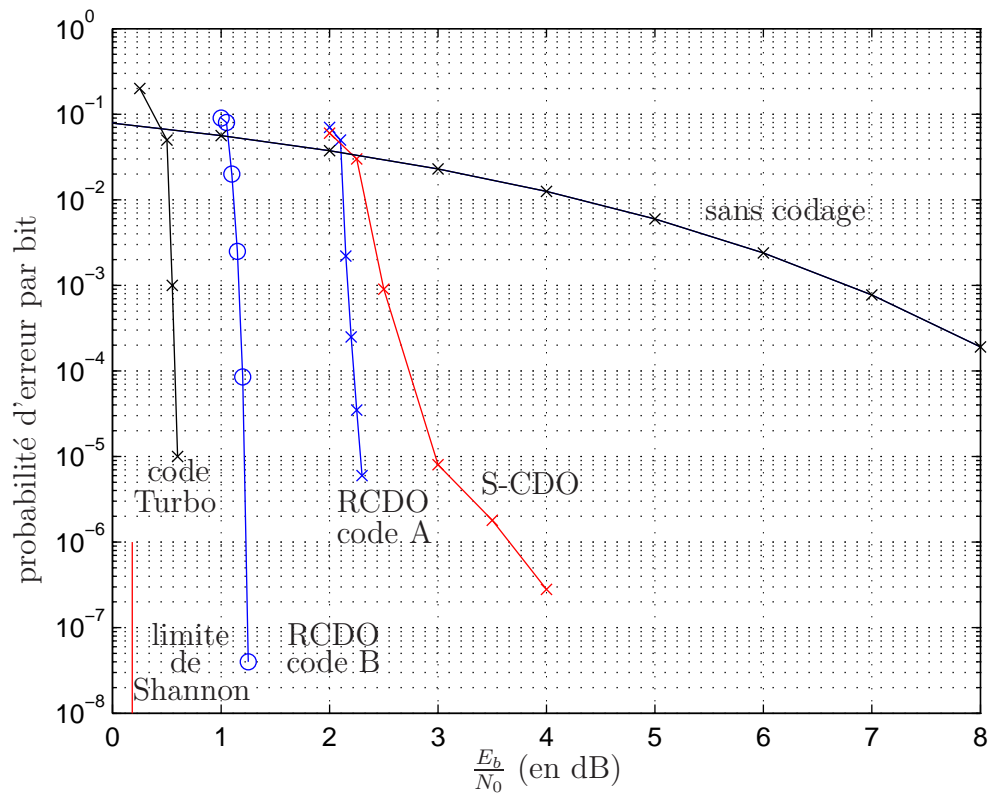


FIGURE 2.7 Comparaison des performances d'erreur entre différentes familles de codes convolutionnels de taux de codage 1/2.

2.4 Codes LDPC

Les codes LDPC en bloc furent introduits par Robert Gallager dans (Gallager, 1962) et sont définis par leur matrice de contrôle qui est creuse. Un code linéaire en bloc (n, k) LDPC (*Low-Density Parity-Check*) est défini à partir de sa matrice de

contrôle \mathbf{H}^T , qui est de faible densité et de dimension $(n - k) \times n$. De façon formelle, un code \mathcal{C} est LDPC si l'ensemble des mots de codes binaires $\mathbf{v} = (v_1, v_2, \dots, v_n)$ représente l'espace nul de \mathbf{H}^T , c'est-à-dire $\mathcal{C} = \{\mathbf{v} \mid \mathbf{v}\mathbf{H}^T = \mathbf{0}, \mathbf{v} \in \mathbb{F}_2^n\}$. Autrement dit, les n symboles binaires formant un mot de code doivent vérifier les $(n - k)$ équations de parité. Un code LDPC est dit régulier si \mathbf{H}^T est composée d'un nombre constant d_λ d'éléments non nuls sur chaque ligne, et possède aussi un nombre constant d_ρ de uns sur les colonnes. Donc chaque symbole v_i , $i = 1, \dots, n$, formant le mot de code \mathbf{v} intervient dans d_λ équations de parité et chaque équation de parité contient d_ρ symboles v_i *différents*.

Un code bloc LDPC régulier peut être représenté par un graphe biparti de Tanner (Tanner, 1981) $G = (V_v \cup V_c, E)$ où E représente l'ensemble des arêtes du graphe, V_v représente l'ensemble des n noeuds désignant les variables (v_1, v_2, \dots, v_n) formant le mot de code, et V_c représente l'ensemble des $(n - k)$ noeuds désignant les contraintes $(c_1, c_2, \dots, c_{n-k})$ qui représentent les $(n - k)$ équations de parité. Une arête ne relie qu'un sommet $v_i \in V_v$ à un sommet $c_j \in V_c$ si et seulement si le symbole v_i intervient dans l'équation de parité représentée par la contrainte c_j , $c_j = \sum_{i=1}^n v_i h_{i,j}$, $j = 1, 2, \dots, (n - k)$. Par conséquent, pour un code régulier tous les sommets de V_v ont un degré d_λ et tous les sommets de V_c ont un degré d_ρ . Il s'en suit que le nombre d'arêtes $|E|$ dans le graphe est donné par :

$$|E| = nd_\lambda \quad (2.4)$$

$$= (n - k)d_\rho \quad (2.5)$$

En combinant (2.4) et (2.5) on peut exprimer le taux de codage $r = k/n$ du code bloc LDPC en fonction des degrés des sommets du graphe biparti :

$$r = 1 - \frac{d_\lambda}{d_\rho} \quad (2.6)$$

La Figure 2.8 montre le graphe biparti général d'un code bloc LDPC régulier où $d_\lambda = 3$ et $d_\rho = 6$. Au centre de la figure, nous retrouvons une boîte noire dans laquelle on retrouve $|E|$ arêtes qui relient selon une certaine configuration les noeuds variables aux noeuds de contraintes. Les noeuds variables, représentés par des cercles, ont tous un degré 3 et les noeuds de contraintes, représentés par des carrés, ont tous un degré 6.

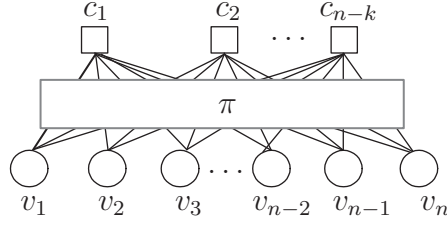


FIGURE 2.8 Graphe biparti d'un code bloc LDPC régulier

Outre les codes blocs LDPC réguliers, nous pouvons aussi considérer la classe des codes blocs LDPC irréguliers. Pour ces codes, les degrés associés aux noeuds variables (contraintes) peuvent varier d'un noeud à l'autre. Pour caractériser un code LDPC en bloc irrégulier (n, k) , nous définissons la paire de distributions $\lambda(x)$ et $\rho(x)$ tel que :

$$\lambda(x) = \sum_{j=2}^{d_\lambda} \lambda_j x^{j-1} \quad \text{et} \quad \rho(x) = \sum_{j=2}^{d_\rho} \rho_j x^{j-1} \quad (2.7)$$

où λ_i représente la fraction des arêtes incidentes aux sommets de degré i dans V_v et ρ_i représente la fraction des arêtes incidentes aux sommets de degré i dans V_c (Richardson *et al.*, 2001). Notons que le degré maximum d'un sommet dans V_v (V_c) est donné par d_λ (d_ρ). En définissant le nombre de sommets de degré i dans V_v (V_c) par n_i^v (n_i^c) nous pouvons écrire les fractions λ_i et ρ_i comme étant :

$$\lambda_j = \frac{j n_j^v}{|E|} \quad \text{et} \quad \rho_j = \frac{j n_j^c}{|E|} \quad (2.8)$$

par conséquent, $\lambda(1) = \rho(1) = \sum_i \lambda_i = \sum_i \rho_i = 1$. À partir de ces distributions, nous pouvons définir le nombre moyen d'arêtes incidentes aux noeuds variables :

$$\overline{d_\lambda} = \frac{|E|}{n} = \frac{1}{\int_0^1 \lambda(x) dx} \quad (2.9)$$

nous pouvons faire de même pour calculer le nombre moyen d'arêtes incidentes sur les noeuds de contrainte :

$$\overline{d_\rho} = \frac{|E|}{(n-k)} = \frac{1}{\int_0^1 \rho(x) dx} \quad (2.10)$$

Nous pouvons ainsi exprimer le taux de codage du code LDPC en bloc irrégulier en fonction des degrés moyens :

$$r = 1 - \frac{\overline{d_\lambda}}{\overline{d_\rho}} \quad (2.11)$$

Nous remarquons que les codes LDPC en bloc réguliers représentent un cas particulier de la classe des codes LDPC en bloc irréguliers. Il a été aussi démontré dans (Luby *et al.*, 2001) et plus récemment dans (Richardson et Urbanke, 2001a) et (Richardson *et al.*, 2001) que seuls les codes en blocs LDPC ayant un graphe biparti irrégulier peuvent approcher la limite de Shannon lorsqu'un algorithme itératif à passage de messages est utilisé (Kschischang *et al.*, 2001).

2.5 Décodage itératif

Généralement, le décodage des codes correcteurs d'erreur s'effectue à l'aide d'un décodeur à maximum de vraisemblance (MLD). Ce type de décodeur est optimum, car il minimise la probabilité d'effectuer une erreur lors du décodage. Pour un code en blocs linéaire (n, k) , un décodeur MLD doit calculer la distance entre le vecteur reçu du canal \mathbf{y} et les 2^{nr} mots de codes possibles. Un décodeur MLD choisira donc le mot de code $\hat{\mathbf{v}} \in \mathcal{C}$ le plus près de \mathbf{y} . Ce faisant, pour décoder un code en bloc avec un décodeur MLD il est nécessaire de comparer 2^{nr} distances. Il s'ensuit que pour un taux de codage r fixé, le nombre de comparaisons devient vite un obstacle lorsque la valeur de n est élevée.

Pour contourner ce problème, Gallager a introduit (Gallager, 1962) deux algorithmes itératifs de décodage qui se basent sur le passage de messages entre les sommets de V_v et de V_c du graphe biparti d'un code LDPC en bloc. Les arêtes de G servent à propager des mesures qui permettent d'estimer les variables. Un message transmis d'un noeud variable v_i vers un noeud de contrainte c_j sera noté $m_{v_i c_j}$ alors que les messages allant d'un noeud de contrainte vers un noeud variable seront notés $m_{c_j v_i}$.

En nous référant à la Figure 2.9, nous pouvons décrire de façon générale le mécanisme des algorithmes itératifs de décodage basés sur la propagation des messages. À l'instant initial $\ell = 0$, Figure 2.9-a), l'algorithme associe à chaque noeud variable v_i du graphe biparti, la valeur $m_{v_i}^{(0)}$ correspondant au message associé à la variable y_i provenant du canal. Ces valeurs sont ensuite propagées vers les noeuds de

contrainte. Pour cet instant uniquement, les messages $m_{v_i c}^{(0)}$ prennent la valeur $m_{v_i}^{(0)}$, $c \in \mathcal{N}(v_i)$. À ce moment, un noeud de contrainte c_j reçoit tous les messages provenant des noeuds variables dans son voisinage $\mathcal{N}(c_j)$. Lorsqu'un noeud de contrainte c_j reçoit les messages $m_{v c_j}^{(0)}$, $v \in \mathcal{N}(c_j)$, l'algorithme retire alors l'information sur la variable $v' \in \mathcal{N}(c_j)$ en utilisant l'information provenant des autres noeuds variables $\mathcal{N}(c_j) \setminus \{v'\}$. Cette information est par la suite retournée vers le noeud v' via le message $m_{c_j v'}^{(0)}$ (Figure 2.9-b). Cette étape est répétée sur l'ensemble des arêtes incidentes au noeud de contrainte c_j et les messages ainsi produits sont transmis vers l'ensemble des noeuds variables $\mathcal{N}(c_j)$, Figure 2.9-c). Par la suite, l'algorithme calcule les messages $m_{v_i c_j}^{(1)}$ en combinant les messages $m_{v_i c'}^{(0)}$, $c' \in \mathcal{N}(v_i) \setminus \{c_j\}$, Figure 2.9-d).

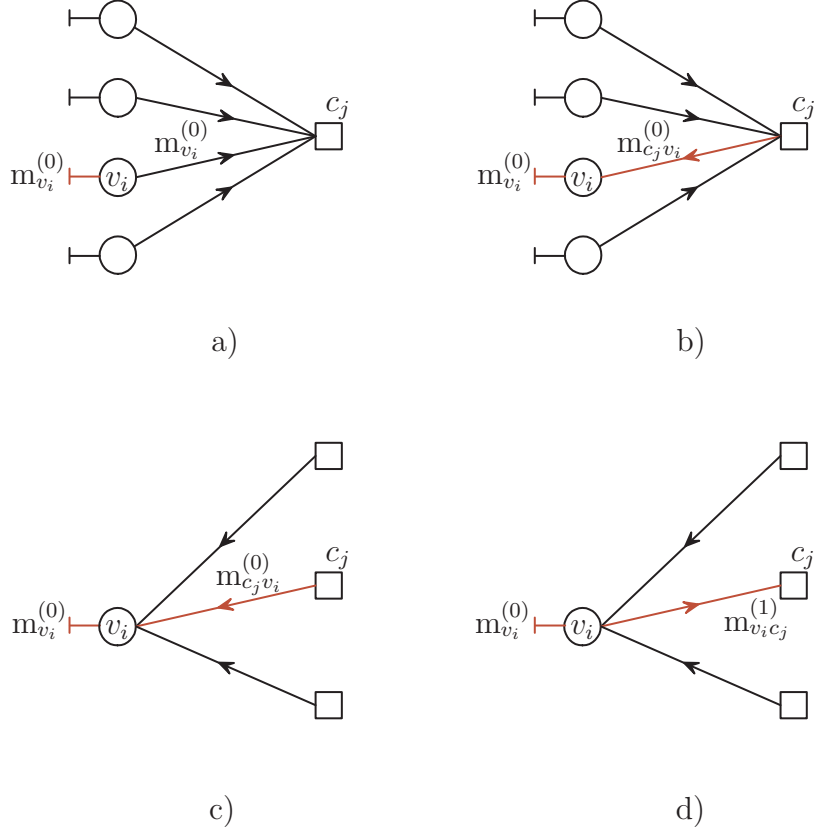


FIGURE 2.9 Échange des messages dans un graphe biparti

En réalité, les messages échangés représentent une mesure basée sur la probabilité *a posteriori* des symboles y_i , ou de façon équivalente sur le logarithme du rapport de vraisemblance (LLR) des symboles y_i . Les règles de mise à jour des messages s'écrivent

donc (Richardson *et al.*, 2001) :

$$m_{v_i c_j}^{(\ell)} = \begin{cases} m_{v_i}^{(0)}, & \text{si } \ell = 0, \\ m_{v_i}^{(0)} + \sum_{c' \in \mathcal{N}(v_i) \setminus \{c_j\}} m_{c' v_i}^{(\ell-1)}, & \text{si } \ell \geq 1, \end{cases} \quad (2.12)$$

où les messages $m_{c' v_i}^{(\ell)}$ s'obtiennent selon la relation suivante :

$$m_{c_j v_i}^{(\ell)} = \ln \frac{1 + \prod_{v' \in \mathcal{N}(c_j) \setminus \{v_i\}} \tanh(m_{v' c_j}^{(\ell)}/2)}{1 - \prod_{v' \in \mathcal{N}(c_j) \setminus \{v_i\}} \tanh(m_{v' c_j}^{(\ell)}/2)} \quad (2.13)$$

L'algorithme associera une valeur binaire dure après avoir effectué \mathcal{L} itérations sur le symbole décodé. Cette décision s'effectue en comparant le message résultant $m_{v_i}^{(\mathcal{L})}$ à un certain seuil. L'organe de décision prend donc la décision dure basée sur la règle suivante :

$$\hat{v}_i = \begin{cases} 0, & \text{si } m_{v_i}^{(\mathcal{L})} \geq 0, \\ 1, & \text{autrement.} \end{cases} \quad (2.14)$$

où

$$m_{v_i}^{(\mathcal{L})} = m_{v_i}^{(0)} + \sum_{c' \in \mathcal{N}(v_i)} m_{c' v_i}^{(\mathcal{L})} \quad (2.15)$$

Nous pouvons constater que le type d'algorithme présenté travaille localement dans le voisinage des noeuds du graphe biparti d'un code en blocs LDPC. À la Section 2.4, nous avons présenté, à la Figure 2.8, le graphe biparti général d'un code en blocs LDPC régulier sans porter attention à la configuration des arêtes dans le graphe. Une question se pose : Se peut-il que la configuration des arêtes dans le graphe ait une incidence sur l'indépendance entre les messages propagés nécessaires à l'estimation des variables ? En fait, pour que l'ensemble des messages propagés soit indépendant, il faut que sur les ℓ itérations effectuées, le graphe biparti se comporte localement comme un arbre au voisinage des noeuds variables (Richardson et Urbanke, 2001a). Autrement dit, il ne devrait pas y avoir de cycles dans le graphe biparti du code en blocs LDPC si on désire que les performances d'erreur de l'algorithme itératif convergent vers celles d'un décodeur MLD.

Trouver un code en blocs LDPC dont le graphe biparti n'a pas de cycles n'est pas facile. Généralement, on admet plusieurs cycles dans le graphe ce qui rend l'algorithme itératif de décodage sous-optimum par rapport au MLD. L'objectif devient

donc d'avoir un graphe biparti dont le plus petit cycle est le plus grand possible. Pour ce faire, il existe plusieurs méthodes de recherche des codes comme par exemple, l'algorithme *Progressive Edge Growth* (PEG) (Sharon et Litsyn, 2006). Cet algorithme permet de placer les arêtes dans le graphe biparti de sorte qu'il ne comporte aucun cycle inférieur à une taille fixée.

À l'origine, Gallager a proposé les codes en blocs LDPC réguliers (d_λ, d_ρ) . Mais est-ce que la régularité des distributions optimise les performances d'erreur du décodeur ? Les auteurs de (Luby *et al.*, 1998) et (Luby *et al.*, 1997) ont répondu à cette question en montrant que si les distributions $\lambda(x)$ et $\rho(x)$ des codes en blocs LDPC irréguliers sont judicieusement choisies, alors l'irrégularité des distributions peut entraîner une amélioration des performances d'erreur des codes en blocs LDPC.

Suite à ces résultats, les auteurs de (Richardson et Urbanke, 2001a) et (Richardson *et al.*, 2001) se sont intéressés à savoir quel est le rapport signal sur bruit E_b/N_0 à partir duquel l'algorithme itératif de décodage peut converger pour une paire de distributions donnée. Autrement dit, sous ce seuil l'algorithme de décodage ne peut atteindre de bonnes performances d'erreur, et ce, même si le nombre d'itérations est très élevé. De plus, les auteurs ont montré qu'on ne peut faire tendre les performances d'erreur vers la limite de Shannon s'il y a des sommets dans le graphe biparti des codes dont le degré est inférieur à deux.

Pour conclure cette section, mentionnons que Sae-Young Chung dans (Chung *et al.*, 2001) a trouvé une paire de distributions qui permet théoriquement de s'approcher à 0.0045 dB de la limite de Shannon pour un code en blocs LDPC irrégulier ayant une longueur de bloc égale à $n = 10^7$ et $r = 1/2$. Ceci se traduit en un problème complexe qui consiste à rechercher un graphe biparti qui ne possède pas de petit cycle, et qui est composé de 10^7 noeuds variables et de 5×10^6 noeuds de contraintes. De plus, pour cette paire de distributions, le degré maximum d_λ d'un noeud variable dans le graphe biparti est égal à 8000. Dans le chapitre suivant, nous effectuons le lien entre les codes convolutionnels récurrents multiregistres doublement orthogonaux (RCDO) et les codes en blocs LDPC. Ceci nous permettra par la suite de déterminer les paires de distributions adaptées aux codes RCDO.

Chapitre 3

Relations entre les codes convolutionnels récurrents RCDO et les codes LDPC

Dans la première partie de ce chapitre, nous montrons que les codes convolutionnels récurrents multiregistres doublement orthogonaux présentés dans (Cardinal, 2001) sont en fait des codes convolutionnels qui se définissent par leur matrice de contrôle creuse. En fait, les codes RCDO représentent une version convolutionnelle des codes blocs LDPC. À partir du lien fait avec les codes blocs LDPC, nous représenterons les codes convolutionnels récurrents multiregistres à l'aide de graphes bipartis fondamentaux, appelées *protographes*. Les protographes représentent la pierre angulaire de cette thèse, car ils nous permettront au prochain chapitre d'observer que la complexité de calculs liée aux codes RCDO dépend uniquement du protographe. De plus au Chapitre 5, les protographes seront utilisés pour établir les limites asymptotiques associées aux codes RCDO.

La seconde partie de ce chapitre nous permet de définir la double orthogonalité des codes RCDO. Comme nous le montrons, l'imposition de conditions sur la position des connexions du codeur convolutionnel permet d'éliminer les petits cycles indésirables dans la structure du graphe biparti des codes RCDO. Cette propriété des codes RCDO permet alors d'effectuer deux itérations consécutives de façon indépendantes les unes des autres.

3.1 Codes convolutionnels récurrents multiregistres RCDO

Comme nous l'avons présenté au chapitre précédent, les codes LDPC en blocs sont générés à partir de leur matrice de contrôle. De la même manière, les codes convolutionnels récurrents doublement orthogonaux sont, eux aussi, définis par leur matrice de vérification. Les codes convolutionnels RCDO sont obtenus à partir de codeurs convolutionnels pour lesquels la position des connexions entre les éléments de délais et les additionneurs modulo 2 ne varient pas dans le temps. Un exemple général d'un codeur convolutionnel récurrent qui génère un code convolutionnel LDPC de taux de codage b/c est présenté à la Figure 3.1, $b < c$ et $b, c \in \mathbb{N}$. Sur cette figure, la valeur $\alpha_{m,n}$ représente la position de la connexion entre le m -ième symbole à la sortie du codeur et le $\alpha_{m,n}$ -ième élément de délai qui constitue le n -ième registre à décalage, $n \in \{1, \dots, (c - b)\}$, $\alpha_{m,n} \in \mathbb{N}$. Pour ce type de codes, à chaque unité

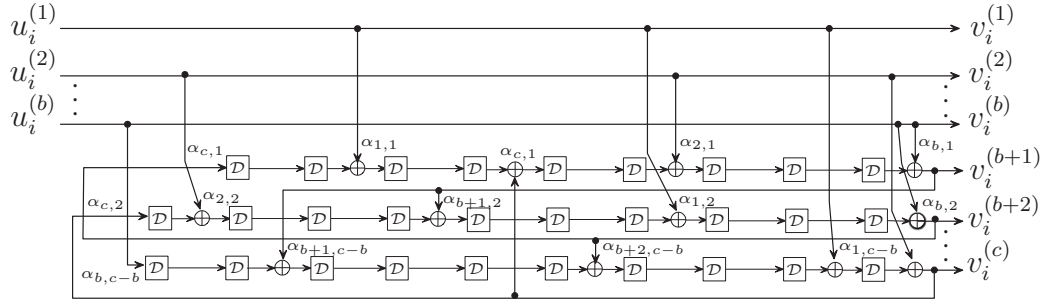


FIGURE 3.1 Exemple d'un codeur convolutionnel récurrent composé de 3 registres en parallèle et d'un taux de codage 3/6.

de temps i , le codeur convolutionnel de taux de codage $r = b/c$ accepte b symboles binaires $u_i^{(1)}, \dots, u_i^{(b)}$ et génère à sa sortie c symboles binaires $v_i^{(1)}, v_i^{(2)}, \dots, v_i^{(c)}$. Nous représentons les c symboles à la sortie du codeur par le vecteur \mathbf{v}_i qui possède c composantes binaires, $\mathbf{v}_i = (v_i^{(1)}, v_i^{(2)}, \dots, v_i^{(c)})$. Le mot de code \mathbf{v} ainsi généré par le codeur convolutionnel est obtenu en recueillant l'ensemble des vecteurs générés à chaque instant i , c'est-à-dire $\mathbf{v} = (\mathbf{v}_0, \mathbf{v}_1, \dots)$.

Les symboles binaires à la sortie du codeur convolutionnel récursif de taux de codage $r = b/c$ sont obtenus selon la relation suivante :

$$v_i^{(m)} = \begin{cases} u_i^{(m)}, & 1 \leq m \leq b; \\ \sum_{\substack{k=1 \\ k \neq m}}^{(c)} v_{i-\alpha_{k,m-b}}^{(k)}, & b+1 \leq m \leq c. \end{cases} \quad (3.1)$$

où $\sum_{\substack{k=1 \\ k \neq m}}^{(c)} v_{i-\alpha_{k,m-b}}^{(k)}$ représente la somme modulo 2 de symboles binaires. L'équation (3.1) nous indique que les b premiers symboles à la sortie du codeur correspondent aux b symboles d'information à l'entrée du codeur convolutionnel, $u_i^{(1)}, \dots, u_i^{(b)}$. Cette propriété rend le code convolutionnel systématique, et est fortement exploitée lors du décodage. Par ailleurs, les $(c-b)$ dernières composantes binaires du vecteur \mathbf{v}_i représentent les $(c-b)$ symboles de parité générés par le codeur. Ces symboles de parité sont obtenus en effectuant la somme modulo 2 entre certaines valeurs binaires qui se retrouvent aux positions $\alpha_{m,n}$ dans les $(c-b)$ registres à décalage en parallèle qui constituent le codeur convolutionnel. Il est important de mentionner que l'équation (3.1) n'est valide que dans le cas où une seule connexion est possible entre le m -ième symbole généré à la sortie du codeur $v_i^{(m)}$ et le n -ième registre à décalage. Autrement dit, un symbole à la sortie du codeur ne peut être injecté plus d'une fois par registre à décalage. Cette propriété permet d'atteindre la double orthogonalité présentée dans (Cardinal, 2001).

Étant donné que les codes convolutionnels récursifs considérés sont systématiques, à chaque instant i , nous pouvons créer un ensemble de $(c-b)$ équations de contraintes. Le schéma fonctionnel de la Figure 3.2 montre comment ces équations de contraintes sont obtenues. Ces contraintes représentent un ensemble d'équations dont la somme modulo 2 est toujours nulle et elles sont définies par l'équation suivante :

$$c_i^{(n)} = v_i^{(b+n)} + \underbrace{\sum_{\substack{m=1 \\ m \neq b+n}}^{(c)} v_{i-\alpha_{m,n}}^{(m)}}_{\text{codage du symbole } \hat{v}_i^{(b+n)}}, \quad 1 \leq n \leq c-b \quad (3.2)$$

Ces équations représentent une comparaison entre le symbole de parité $v_i^{(b+n)}$ généré par l'équation (3.1) et le ré-encodage de ce dernier symbole via la partie systématique du premier codeur convolutionnel. Nous pouvons représenter l'ensemble des contraintes

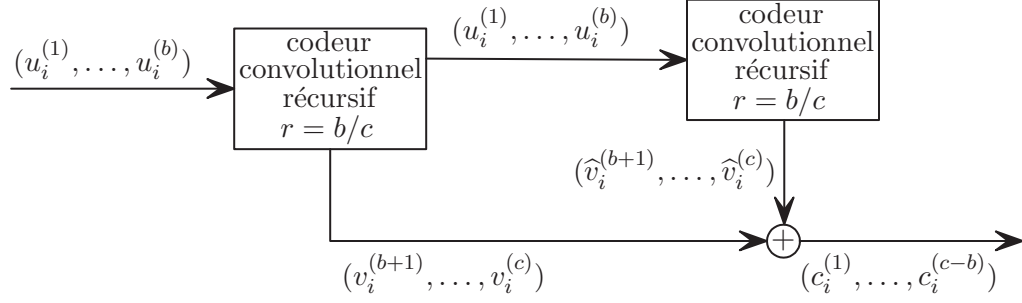


FIGURE 3.2 Schéma fonctionnel représentant la formation des équations de contraintes.

à l'instant i par le vecteur $\mathbf{c}_i = (c_i^{(1)}, \dots, c_i^{(c-b)})$. Ce faisant l'ensemble des contraintes associées au mot de code \mathbf{v} peut s'écrire $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2, \dots)$.

Le mot de code \mathbf{v} généré par un codeur RCDO est donc valide lorsque l'ensemble des équations de contraintes est vérifié, c'est-à-dire que la somme modulo 2 de chaque équation de contraintes est égale à zéro. Nous pouvons écrire ce système d'équations sous la forme matricielle suivante :

$$\mathbf{c} = \mathbf{v}\mathbf{H}^T \quad (3.3)$$

$$= \mathbf{0} \quad (3.4)$$

où la matrice \mathbf{H}^T représente la matrice de contrôle semi-infinie dont les éléments sont binaires. Un mot de code représente donc l'espace nul de la matrice \mathbf{H}^T . C'est cette matrice de vérification qui définit un code convolutionnel récursif LDPC. Les dimensions de la matrice binaire \mathbf{H}^T font en sorte qu'il devient difficile d'exprimer les propriétés d'un code convolutionnel récursif LDPC à partir de sa matrice \mathbf{H}^T . Cependant, comme un codeur convolutionnel est un système linéaire, nous pouvons exprimer l'opération de convolution dans le temps à l'aide de la transformée en D , (Lin et Costello, 2004). Sous cette transformation, l'opération de convolution est représentée par une multiplication de polynômes et permet une représentation de la matrice \mathbf{H}^T sous une forme plus compacte. Avec cette représentation, le mot de code généré par le codeur peut s'écrire $\mathbf{v}(D)$, $\mathbf{v}(D) = (v^{(1)}(D), \dots, v^{(c)}(D))$. Sous cette forme, la séquence binaire associée à la m -ième sortie du codeur convolutionnel s'écrit

sous la forme polynomiale suivante :

$$v^{(m)}(D) = \sum_i v_i^{(m)} D^i, \quad v_i^{(m)} \in \{0, 1\} \quad (3.5)$$

où $v_i^{(m)}$ représente le symbole binaire à la m -ième sortie du codeur convolutionnel à l'instant i . Sous cette transformation, la matrice de vérification binaire \mathbf{H}^T peut être exprimée par la matrice de contrôle $\mathbf{H}^T(D)$ suivante :

$$\mathbf{H}^T(D) = \begin{pmatrix} D^{\alpha_{1,1}} & \dots & D^{\alpha_{1,(c-b)}} \\ D^{\alpha_{2,1}} & \dots & D^{\alpha_{2,(c-b)}} \\ \vdots & & \vdots \\ D^{\alpha_{b,1}} & \dots & D^{\alpha_{b,(c-b)}} \\ D^{\alpha_{b+1,1}} & \dots & D^{\alpha_{b+1,(c-b)}} \\ \vdots & & \vdots \\ D^{\alpha_{c,1}} & \dots & D^{\alpha_{c,(c-b)}} \end{pmatrix} \quad (3.6)$$

de dimension $c \times (c - b)$. En utilisant cette représentation, nous pouvons écrire le vecteur de contraintes $\mathbf{c}(D)$ qui est égal à $(c^{(1)}(D), \dots, c^{(c-b)}(D))$ où,

$$c^{(n)}(D) = \sum_i c_i^{(n)} D^i, \quad c_i^{(n)} \in \{0, 1\} \quad (3.7)$$

et l'équation (3.3) devient,

$$\mathbf{c}(D) = \mathbf{v}(D) \mathbf{H}^T(D) \quad (3.8)$$

En effectuant la multiplication matricielle (3.8), l'équation (3.7) devient :

$$c^{(n)}(D) = \sum_i \sum_{m=1}^c v_i^{(m)} D^{i+\alpha_{m,n}} \quad (3.9)$$

de sorte qu'en égalisant le terme en D^i du polynôme représenté par l'équation (3.7)

avec les termes en D^i de l'équation (3.9), nous observons :

$$\begin{aligned}
c_i^{(n)} &= \sum_{m=1}^c v_{i-\alpha_{m,n}}^{(m)} \\
&= v_{i-\alpha_{b+n,n}}^{(b+n)} + \sum_{\substack{m=1 \\ m \neq b+n}}^c v_{i-\alpha_{m,n}}^{(m)}
\end{aligned} \tag{3.10}$$

Nous pouvons remarquer que l'équation (3.10) représente bien la relation présentée à l'équation (3.2) lorsque la valeur $\alpha_{b+n,n}$ est égale à zéro. Comme les valeurs $\alpha_{b+n,n}$ de (3.10) doivent prendre la valeur zéro, il s'ensuit que les éléments formant la diagonale inférieure de la matrice $\mathbf{H}^T(D)$ sont égaux à 1, $1 \leq n \leq (c-b)$. Remarquons que la taille de la matrice de vérification $\mathbf{H}^T(D)$ dépend uniquement du taux de codage r du code convolutionnel récursif, $r = b/c$, $b < c$ et par conséquent, du nombre de registres à décalage $(c-b)$. Sous cette transformation, la matrice de parité $\mathbf{H}^T(D)$ nous permet donc de réécrire le système d'équations défini par l'équation (3.3) sous la forme suivante :

$$\begin{aligned}
\mathbf{c}(D) &= \mathbf{v}(D)\mathbf{H}^T(D) \\
&= \mathbf{0}(D)
\end{aligned} \tag{3.11}$$

où $\mathbf{0}(D)$ représente le vecteur de polynômes nuls. C'est-à-dire que toutes les équations de contraintes $c^{(n)}(D)$ représentées par (3.7) sont égales au polynôme nul $\mathbf{0}(D)$, où $\mathbf{0}(D) = 0 + 0D + 0D^2 + \dots$.

L'équation (3.6) représente le coeur même de cette thèse, à partir de cette matrice, nous associerons plusieurs caractéristiques aux codes convolutionnels récursifs LDPC à temps-invariant. Entre autres, c'est en identifiant certaines conditions sur les valeurs $\alpha_{m,n}$ que nous pouvons définir les conditions de la double orthogonalité des codes convolutionnels récursifs RCDO. De plus, comme nous le montrons au Chapitre 5, c'est en contrôlant le nombre d'éléments non nuls dans la matrice de l'équation (3.6) que nous serons en mesure d'établir que certaines configurations de ces éléments permettent d'approcher les performances d'erreur des codes RCDO de la capacité de Shannon.

Exemple 3.1— Prenons par exemple un code convolutionnel récursif de taux de codage $r = 3/6$ qui possède 3 registres à décalage en parallèle. Ce code peut être généré à l'aide d'un codeur convolutionnel comme celui présenté à la Figure 3.3.

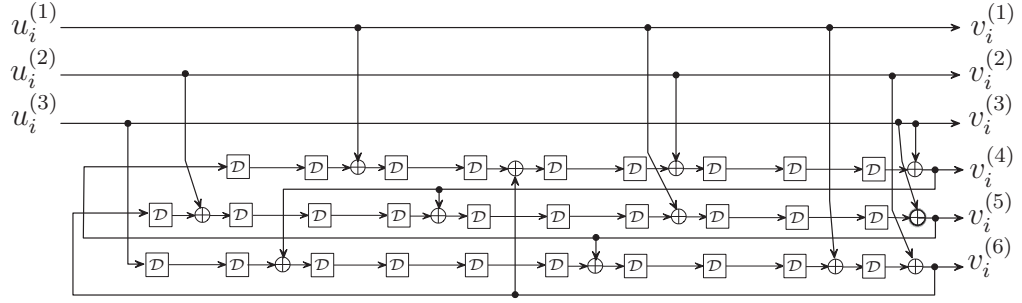


FIGURE 3.3 Exemple d'un codeur convolutionnel récursif composé de 3 registres en parallèle et d'un taux de codage 3/6.

Selon l'équation (3.1), à chaque instant i , un codeur convolutionnel récursif systématique produit la partie systématique du code qui correspond aux trois symboles binaires $v_i^{(m)}$, $1 \leq m \leq 3$, qui prennent les valeurs suivantes :

$$\begin{aligned} v_i^{(1)} &= u_i^{(1)} \\ v_i^{(2)} &= u_i^{(2)} \\ v_i^{(3)} &= u_i^{(3)} \end{aligned}$$

Au même instant, le codeur produit aussi trois symboles de parité $v_i^{(m)}$, $4 \leq m \leq 6$, qui prennent comme valeurs :

$$v_i^{(4)} = v_{i-\alpha_{1,1}}^{(1)} + v_{i-\alpha_{2,1}}^{(2)} + v_{i-\alpha_{3,1}}^{(3)} + v_{i-\alpha_{5,1}}^{(5)} + v_{i-\alpha_{6,1}}^{(6)} \quad (3.12)$$

$$v_i^{(5)} = v_{i-\alpha_{1,2}}^{(1)} + v_{i-\alpha_{2,2}}^{(2)} + v_{i-\alpha_{3,2}}^{(3)} + v_{i-\alpha_{4,2}}^{(4)} + v_{i-\alpha_{6,2}}^{(6)} \quad (3.13)$$

$$v_i^{(6)} = v_{i-\alpha_{1,3}}^{(1)} + v_{i-\alpha_{2,3}}^{(2)} + v_{i-\alpha_{3,3}}^{(3)} + v_{i-\alpha_{4,3}}^{(4)} + v_{i-\alpha_{5,3}}^{(5)} \quad (3.14)$$

Les équations (3.12) à (3.14) représentent les équations de parité, sous leur forme générale, qui sont utilisées par un codeur convolutionnel récursif pour générer un code convolutionnel systématique de taux de codage

$$r = 3/6.$$

Pour le codeur convolutionnel récursif présenté à la Figure 3.3, où les éléments de délais qui constituent les registres à décalage sont représentés par le symbole \mathcal{D} , nous pouvons réécrire les équations (3.12) à (3.14) en remplaçant les valeurs $\alpha_{m,n}$ par leur valeur respective. Ce qui nous conduit aux équations :

$$v_i^{(4)} = v_{i-7}^{(1)} + v_{i-3}^{(2)} + v_i^{(3)} + v_{i-9}^{(5)} + v_{i-5}^{(6)} \quad (3.15)$$

$$v_i^{(5)} = v_{i-3}^{(1)} + v_{i-9}^{(2)} + v_i^{(3)} + v_{i-6}^{(4)} + v_{i-10}^{(6)} \quad (3.16)$$

$$v_i^{(6)} = v_{i-1}^{(1)} + v_i^{(2)} + v_{i-10}^{(3)} + v_{i-8}^{(4)} + v_{i-4}^{(5)} \quad (3.17)$$

Par exemple, la valeur $\alpha_{5,3}$ de l'équation (3.14) prend la valeur 4 dans l'équation (3.17), car la connexion entre la cinquième valeur en sortie $v_i^{(5)}$ et le troisième registre à décalage se situe à la suite du quatrième élément de délai de ce registre à décalage.

De plus, les trois équations de contraintes que nous pouvons associer au code généré s'écrivent :

$$c_i^{(1)} = v_{i-7}^{(1)} + v_{i-3}^{(2)} + v_i^{(3)} + v_i^{(4)} + v_{i-9}^{(5)} + v_{i-5}^{(6)} \quad (3.18)$$

$$c_i^{(2)} = v_{i-3}^{(1)} + v_{i-9}^{(2)} + v_i^{(3)} + v_{i-6}^{(4)} + v_i^{(5)} + v_{i-10}^{(6)} \quad (3.19)$$

$$c_i^{(3)} = v_{i-1}^{(1)} + v_i^{(2)} + v_{i-10}^{(3)} + v_{i-8}^{(4)} + v_{i-4}^{(5)} + v_i^{(6)} \quad (3.20)$$

À partir des équations de contraintes, nous pouvons déduire que la matrice de vérification $\mathbf{H}^T(D)$ associée à ce code est représentée par l'équation (3.21).

$$\mathbf{H}^T(D) = \begin{pmatrix} D^7 & D^3 & D \\ D^3 & D^9 & 1 \\ 1 & 1 & D^{10} \\ 1 & D^6 & D^8 \\ D^9 & 1 & D^4 \\ D^5 & D^{10} & 1 \end{pmatrix} \quad (3.21)$$

3.2 Protographes des codes convolutionnels récurrents multiregistres

La matrice de contrôle $\mathbf{H}^T(D)$ des codes convolutionnels multiregistres RCDO dans l'équation (3.6) représente une version abstraite d'un code convolutionnel. À partir de cette matrice, il devient difficile d'identifier les paramètres d'intérêts qui caractérisent les performances d'un code. Cependant, en considérant une représentation graphique de la matrice de vérification à l'aide d'un graphe biparti il devient simple de définir les conditions de double orthogonalité. En 1981, Tanner a montré qu'il est possible de représenter la relation qui existe entre les variables et les équations de contraintes du code à l'aide d'un graphe biparti (Tanner, 1981). Ces graphes bipartis sont appelés graphes de Tanner et certains d'entre eux peuvent être obtenus à partir de plus petit graphes bipartis fondamentaux, appelés *protographes* (Thorpe, 2003). À l'aide de ces protographes, nous établirons au chapitre cinq, la relation entre les performances asymptotiques à faible rapport signal sur bruit des codes RCDO, et la répartition des éléments non nuls dans la matrice de contrôle $\mathbf{H}^T(D)$. De plus, au prochain chapitre nous établirons facilement à partir des protographes la complexité associée au décodage et à l'encodage des codes convolutionnels RCDO.

Nous rappelons la définition de la matrice de contrôle $\mathbf{H}^T(D)$ d'un code RCDO de taux de codage $r = b/c$ sous une forme plus générale :

$$\mathbf{H}^T(D) = \begin{pmatrix} h_{1,1}D^{\alpha_{1,1}} & \dots & h_{1,(c-b)}D^{\alpha_{1,(c-b)}} \\ h_{2,1}D^{\alpha_{2,1}} & \dots & h_{2,(c-b)}D^{\alpha_{2,(c-b)}} \\ \vdots & & \vdots \\ h_{b,1}D^{\alpha_{b,1}} & \dots & h_{b,(c-b)}D^{\alpha_{b,(c-b)}} \\ h_{b+1,1}D^{\alpha_{b+1,1}} & \dots & h_{b+1,(c-b)}D^{\alpha_{b+1,(c-b)}} \\ \vdots & & \vdots \\ h_{c,1}D^{\alpha_{c,1}} & \dots & h_{c,(c-b)}D^{\alpha_{c,(c-b)}} \end{pmatrix} \quad (3.22)$$

où nous avons introduit les coefficients $h_{m,n}, h_{m,n} \in \{0, 1\}$, pour représenter l'existence d'une connexion entre le m -ième symbole à la sortie du codeur convolutionnel et le n -ième registre à décalage. La valeur $h_{m,n}$ prend la valeur un lorsqu'une connexion existe entre le m -ième symbole à la sortie du codeur RCDO et le n -ième registre à décalage. Autrement, la valeur $h_{m,n}$ est égale à zéro.

Définition 3.1 : Un protographe $G_{\mathcal{P}} = (V_v^{\mathcal{P}} \cup V_c^{\mathcal{P}}, E_{\mathcal{P}})$ est un graphe biparti constitué d'un ensemble d'arêtes $E_{\mathcal{P}}$ qui relie les sommets de l'ensemble $V_v^{\mathcal{P}}$ aux sommets de l'ensemble $V_c^{\mathcal{P}}$.

Nous associons un protographe $G_{\mathcal{P}}$ à une matrice de vérification $\mathbf{H}^T(D)$ de la façon suivante, chaque ligne de $\mathbf{H}^T(D)$ représente un noeud de $V_v^{\mathcal{P}}$ et chaque noeud de l'ensemble $V_c^{\mathcal{P}}$ représente une colonne de $\mathbf{H}^T(D)$. L'ensemble des noeuds $V_v^{\mathcal{P}}$ se nomme l'ensemble de noeuds variables et le nombre de noeuds dans cet ensemble $|V_v^{\mathcal{P}}|$ est égal à c noeuds, car la matrice $\mathbf{H}^T(D)$ est composée de c lignes. De la même manière, nous notons l'ensemble des noeuds de contraintes $V_c^{\mathcal{P}}$ qui contient un nombre de noeuds $|V_c^{\mathcal{P}}|$ égal à $(c - b)$ noeuds qui correspondent aux $(c - b)$ colonnes de la matrice $\mathbf{H}^T(D)$. Un noeud de l'ensemble $V_v^{\mathcal{P}}$ est relié à un noeud de l'ensemble $V_c^{\mathcal{P}}$ si et seulement si la valeur $h_{m,n}$ est égale à 1, c'est-à-dire que l'intersection entre la m -ième ligne et la n -ième colonne de $\mathbf{H}^T(D)$ n'est pas un élément non nul. Le nombre d'arêtes $|E_{\mathcal{P}}|$ dans l'ensemble $E_{\mathcal{P}}$ est donc égal aux nombres d'éléments non nuls dans la matrice de vérification $\mathbf{H}^T(D)$. De plus, nous noterons $d(v^{(m)})$ le degré du m -ième noeud de l'ensemble $V_v^{\mathcal{P}}$. Le degré d'un noeud représente le nombre d'arêtes incidentes au noeud $v^{(m)}$ dans le protographe, $m \in \{1, \dots, c\}$. Cette valeur est donc égale au nombre d'éléments non nuls à la m -ième ligne de la matrice $\mathbf{H}^T(D)$. De la même manière, le degré $d(c^{(n)})$ représente le nombre d'arêtes incidentes au noeud $c^{(n)}$ de l'ensemble $V_c^{\mathcal{P}}$, et correspond au nombre d'éléments non nul de la n -ième colonne de $\mathbf{H}^T(D)$. On définit le degré maximum d'un noeud variable dans $G_{\mathcal{P}}$ par d_{λ} :

$$d_{\lambda} = \max_{1 \leq m \leq b} (d(v^{(m)}))$$

et le degré maximum d'un noeud de contraintes par d_{ρ} :

$$d_{\rho} = \max_{1 \leq n \leq (c-b)} (d(c^{(n)}))$$

La Figure 3.4 présente un protographe général associé à un ensemble de codes convolutionnels récurrents LDPC qui utilisent plusieurs registres à décalage en parallèle. Sur cette figure, les noeuds de contraintes $c^{(n)}$ de l'ensemble $V_c^{\mathcal{P}}$ sont représentés par des carrés tandis que les noeuds variables $v^{(m)}$ de $V_v^{\mathcal{P}}$ sont représentés par des cercles. Cette figure met en évidence deux types de noeuds variables : ceux associés aux symboles d'information $v^{(m)}$, $1 \leq m \leq b$, et ceux qui représentent les symboles de

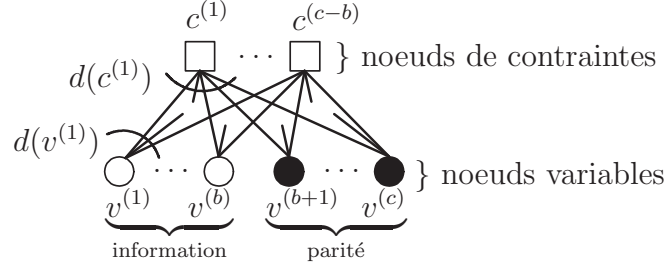


FIGURE 3.4 Protographe général $G_{\mathcal{P}}$ d'un ensemble de code convolutionnels récurrents multi-registres de taux de codage $= b/c$.

parité $v^{(m)}$, $b + 1 \leq m \leq c$. Les noeuds représentant les symboles d'information sont représentés par un cercle blanc et les noeuds correspondant aux symboles de parité sont représentés par des cercles noirs.

Définition 3.2 : Un graphe non orienté $G = (V, E)$ est dit *connexe* si quels que soient les sommets u et v de V il existe une chaîne de u vers v .

Définition 3.3 : Un protographe connexe $G_{\mathcal{P}} = (V_v^{\mathcal{P}} \cup V_c^{\mathcal{P}}, E_{\mathcal{P}})$ est dit *régulier* si tous les noeuds de l'ensemble $V_v^{\mathcal{P}}$ ont leur degré égal à d_{λ} et tous les noeuds de $V_c^{\mathcal{P}}$ ont leur degré égal à d_{ρ} .

Définition 3.4 : Un protographe connexe $G_{\mathcal{P}} = (V_v^{\mathcal{P}} \cup V_c^{\mathcal{P}}, E_{\mathcal{P}})$ est dit *irrégulier* si $G_{\mathcal{P}}$ n'est pas régulier.

Pour alléger le reste du document, lorsque nous parlons de graphes, nous assumons implicitement qu'ils sont connexes. De plus, pour le reste de la thèse, nous associerons à un protographe certains paramètres. Les paramètres d'intérêt sont : –les nombres de sommets dans les ensembles $V_v^{\mathcal{P}}$ et $V_c^{\mathcal{P}}$, –le nombre d'arêtes $|E_{\mathcal{P}}|$ qui le constitue, –ainsi que les fractions associées à l'incidence des arêtes aux noeuds variables et de contraintes. Ces fractions sont représentées à l'aide de deux fonctions d'énumération. L'une, $\lambda(x)$, correspond aux fractions des arêtes incidentes aux noeuds variables d'un certain degré. Et l'autre, $\rho(x)$, correspond aux fractions des arêtes incidentes aux noeuds de contraintes. Un protographe sera donc représenté par son nombre d'arêtes ainsi que par la paire de fonctions $(\lambda(x), \rho(x))$ définies par :

$$\lambda(x) = \sum_{j=2}^{d_{\lambda}} \lambda_j x^{j-1} \quad \text{et} \quad \rho(x) = \sum_{j=2}^{d_{\rho}} \rho_j x^{j-1} \quad (3.23)$$

où la valeur λ_j (ρ_j) correspond à la fraction des arêtes incidentes à des noeuds variables (contrainte) de degré j de l'ensemble $V_v^{\mathcal{P}}$ ($V_c^{\mathcal{P}}$). Les fractions λ_j et ρ_j s'expriment par :

$$\lambda_j = \frac{jn_j^v}{|E_{\mathcal{P}}|} \quad \text{et} \quad \rho_j = \frac{jn_j^c}{|E_{\mathcal{P}}|} \quad (3.24)$$

où n_j^v et n_j^c représentent respectivement le nombre de noeuds variables et de contraintes de $V_v^{\mathcal{P}}$ et $V_c^{\mathcal{P}}$ de degré j . Par conséquent, les valeurs entières n_j^v et n_j^c représentent respectivement, le nombre de lignes et de colonnes de $\mathbf{H}^T(D)$ qui possèdent j valeurs différentes de zéro. Pour simplifier la notation, nous noterons la paire de distributions $(\lambda(x), \rho(x))$ simplement (λ, ρ) .

Notons que le protographe peut être vu comme étant une version non temporelle des équations de contraintes décrites par l'équation (3.2). Sans tenir compte de l'indice temporel des différentes variables qui apparaissent dans (3.2), nous pouvons écrire les équations de contraintes non temporelles suivantes :

$$c^{(n)} = v^{(b+n)} + \sum_{\substack{k=1 \\ k \neq b+n}}^{(c)} v^{(k)} \quad 1 \leq n \leq c - b \quad (3.25)$$

C'est-à-dire que le protographe représente la relation entre les c différentes variables à la sortie du codeur $v^{(m)}$ et leurs apparitions dans les $(c - b)$ équations de contraintes non temporelles $c^{(n)}$. Étant donné qu'une équation de contraintes correspond aussi à un registre à décalage, il s'ensuit que le protographe nous indique aussi dans quel registre à décalage les variables $v^{(m)}$ sont injectées. Par conséquent, une variable $v^{(m)}$ qui apparaît dans k équations de contraintes non temporelles se voit représentée par un noeud variable dans $G_{\mathcal{P}}$ qui possède un degré $d(v^{(m)})$ égal à k dans le protographe. Les k arêtes incidentes au noeud $v^{(m)}$ sont aussi incidentes à k noeuds de contraintes $c^{(n)}$ différents dans l'ensemble $V_c^{\mathcal{P}}$. Ceci revient à dire que la variable à la m -ième sortie du codeur est injectée dans k registres à décalage différents.

Exemple 3.2— Pour cet exemple nous montrons que le protographe, qui caractérise le code convolutionnel récursif LDPC de la Figure 3.3 de l'exemple 3.1 est régulier. Rappelons que la matrice de vérification pour

ce code est exprimée par :

$$\mathbf{H}^T(D) = \begin{pmatrix} D^7 & D^3 & D \\ D^3 & D^9 & 1 \\ 1 & 1 & D^{10} \\ 1 & D^6 & D^8 \\ D^9 & 1 & D^4 \\ D^5 & D^{10} & 1 \end{pmatrix}$$

Pour ce code, les équations de contraintes s'écrivent selon les équations (3.18) à (3.20). Par conséquent, les équations de contraintes non temporelles peuvent s'écrire :

$$c^{(1)} = v^{(4)} + v^{(1)} + v^{(2)} + v^{(3)} + v^{(5)} + v^{(6)} \quad (3.26)$$

$$c^{(2)} = v^{(5)} + v^{(1)} + v^{(2)} + v^{(3)} + v^{(4)} + v^{(6)} \quad (3.27)$$

$$c^{(3)} = v^{(6)} + v^{(1)} + v^{(2)} + v^{(3)} + v^{(4)} + v^{(5)} \quad (3.28)$$

Le protographe $G_{\mathcal{P}}$ correspondant à ce code est donc défini par le graphe biparti de la Figure 3.5.

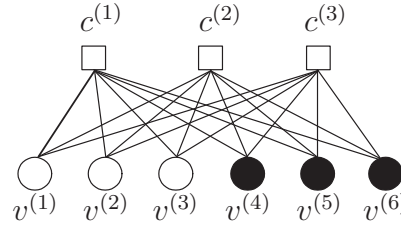


FIGURE 3.5 Protographe *régulier* qui représente le code convolutionnel récursif de taux de codage $= 3/6$ de la Figure 3.3.

Comme nous pouvons le constater, chaque équation de contraintes fait intervenir six variables. De ce fait, chaque nœud de contraintes dans le protographe possède un degré égal à six. De plus, chacune des variables à la sortie du codeur apparaît dans les trois équations de contraintes, par conséquent chaque nœud variable possède trois arêtes qui lui sont incidentes. Étant donné que tous les nœuds variables possèdent un degré d_{λ} égal à 3 et que chaque nœud de contraintes possède un degré d_{ρ} égal

à 6, $G_{\mathcal{P}}$ est défini comme étant *régulier*. De plus, ce protographe régulier possède un nombre total d'arêtes $|E_{\mathcal{P}}|$ égal à 18, car $\mathbf{H}^T(D)$ est composée de 18 éléments non nul.

La paire de distributions des arêtes (λ, ρ) pour ce protographe est donc (x^2, x^5) , car tous les noeuds variables sont de degré trois et tous les noeuds de contraintes sont de degré six. Remarquons que les paires de distributions régulières du type $(x^{d_\lambda-1}, x^{d_\rho-1})$ sont complètement caractérisées par les valeurs d_λ et d_ρ . Par conséquent, pour le reste de ce document, les paires de distribution régulières (λ, ρ) seront représentées soit par le polynôme $(x^{d_\lambda-1}, x^{d_\rho-1})$, où bien par la paire de degrés (d_λ, d_ρ) . Il s'ensuit que le protographe de la Figure 3.5 peut aussi être caractérisé par la paire de degrés $(3, 6)$.

Exemple 3.3— Dans cet exemple, nous considérons le code convolutionnel LDPC de taux de codage 3/6 qui est généré par le codeur convolutionnel de la Figure 3.6. Ce codeur convolutionnel est semblable au codeur convolutionnel présenté aux exemples 3.1 et 3.2, mais pour celui-ci nous avons éliminé trois connexions qui relient les symboles générés par le codeur à certains registres à décalage.

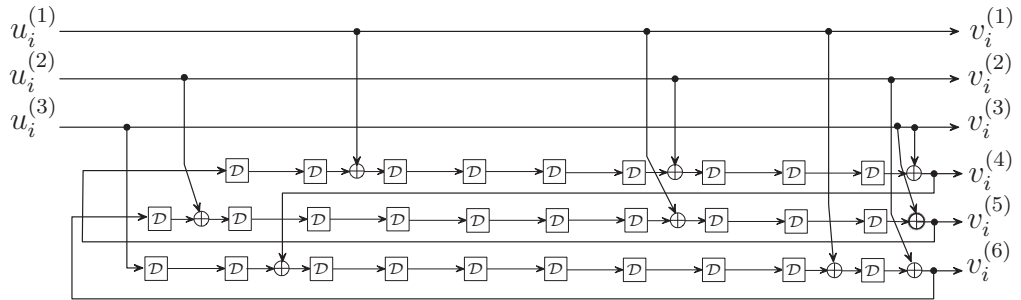


FIGURE 3.6 Exemple d'un codeur convolutionnel récursif composé de 3 registres en parallèle et d'un taux de codage $r = 3/6$.

Les connexions éliminées sont : celle entre le sixième symbole à la sortie et le premier registre à décalage, celle entre le quatrième symbole en sortie et le deuxième registre à décalage et celle entre le cinquième symbole à la

sortie et le troisième registre à décalage. Il s'ensuit que les valeurs $h_{m,n}$ de la matrice de vérification $\mathbf{H}^T(D)$ prennent toutes la valeur 1 à l'exception de $h_{6,1}$, $h_{4,2}$ et $h_{5,3}$ qui prennent la valeur 0. C'est-à-dire que la matrice de vérification pour ce code est donnée par :

$$\mathbf{H}^T(D) = \begin{pmatrix} D^7 & D^3 & D \\ D^3 & D^9 & 1 \\ 1 & 1 & D^{10} \\ 1 & 0 & D^8 \\ D^9 & 1 & 0 \\ 0 & D^{10} & 1 \end{pmatrix} \quad (3.29)$$

Les équations de contraintes intemporelles s'écrivent donc de la façon suivante :

$$c^{(1)} = v^{(1)} + v^{(2)} + v^{(3)} + v^{(4)} + v^{(5)} \quad (3.30)$$

$$c^{(2)} = v^{(1)} + v^{(2)} + v^{(3)} + v^{(5)} + v^{(6)} \quad (3.31)$$

$$c^{(3)} = v^{(1)} + v^{(2)} + v^{(3)} + v^{(4)} + v^{(6)} \quad (3.32)$$

Chaque équation de contraintes intemporelle pour ce code est donc représentée dans le protographe de la Figure 3.7 par un noeud de degré égal à cinq, car chaque équation de contraintes contient cinq variables.

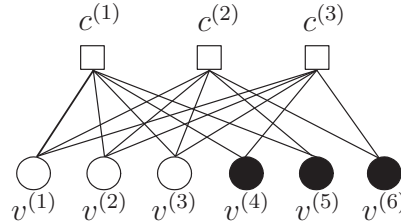


FIGURE 3.7 Protographe *irrégulier* représentant le code convolutionnel récursif de taux de codage $r = 3/6$ de la Figure 3.6.

En comparaison avec l'exemple précédent, dans cet exemple les noeuds variables du protographe ne possèdent pas tous un degré constant, car ils n'apparaissent pas le même nombre de fois au sein des équations de contraintes. Par exemple, la variable en sortie $v^{(1)}$ apparaît dans les trois

équations de contraintes, mais la variable $v^{(4)}$ n'apparaît que deux fois dans l'ensemble des équations de contraintes. Le protographe correspondant à ce code convolutionnel est donc *irrégulier* et possède un nombre total d'arêtes égal à 15, car $\mathbf{H}^T(D)$ possède 15 éléments non nuls. Comme il y a six arêtes incidentes à trois noeuds variables de degré égal à deux et neuf arêtes incidentes à trois noeuds variables de degré trois, il s'ensuit que $\lambda_2 = 6/15$ et $\lambda_3 = 9/15$. Comme tous les noeuds de contraintes du protographe possèdent un degré égal à cinq, $\rho_5 = 1$. La paire de distributions (λ, ρ) est donc,

$$\lambda(x) = \frac{6}{15}x + \frac{9}{15}x^2 \quad \text{et} \quad \rho(x) = x^4 \quad (3.33)$$

Notons qu'il existe plusieurs protographes qui peuvent correspondre à une paire de distributions (λ, ρ) . C'est pour cette raison qu'on associe un ensemble de codes convolutionnels récurrents à une paire de distributions.

Remarque : Jusqu'à présent, nous avons présenté le protographe des codes convolutionnels récurrents de taux de codage $r = b/c$. Cependant, un cas particulier de la matrice de vérification des codes présentés est celui où les éléments de la partie inférieure de la matrice $\mathbf{H}^T(D)$ sont tous égaux à zéro à l'exception de la diagonale inférieure dont les éléments $\alpha_{(b+n),n}$ prennent la valeur zéro, et où $h_{(b+n),n}$ est égal un. La matrice de vérification pour ce cas de figure est donc celle décrite par l'équation (3.34).

$$\mathbf{H}^T(D) = \begin{pmatrix} h_{1,1}D^{\alpha_{1,1}} & \dots & h_{1,(c-b)}D^{\alpha_{1,(c-b)}} \\ h_{2,1}D^{\alpha_{2,1}} & \dots & h_{2,(c-b)}D^{\alpha_{2,(c-b)}} \\ \vdots & & \vdots \\ h_{b,1}D^{\alpha_{b,1}} & \dots & h_{b,(c-b)}D^{\alpha_{b,(c-b)}} \\ 1 & 0 \dots & 0 \\ 0 & 1 & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots 0 & 1 \end{pmatrix} \quad (3.34)$$

Ce cas de figure représente en fait les codes convolutionnels non récursifs qui peuvent être générés par un codeur convolutionnel qui utilise plusieurs registres à décalage en parallèle. Un exemple de ce type de codeur est présenté à la Figure 2.5. Pour ce cas particulier, nous pouvons remarquer que tous les noeuds variables $v_i^{(b+n)}$, représentant les symboles de parité, sont de degré égal à un, car les $(c - b)$ dernières lignes de la matrice $\mathbf{H}^T(D)$ contiennent qu'un seul élément non nul. Nous pouvons donc représenter ces codes par le protographe de la Figure 3.8. Ce protographe irrégulier correspond en fait à celui qui est associé aux codes convolutionnels doublement orthogonaux définis dans (Cardinal, 2001).

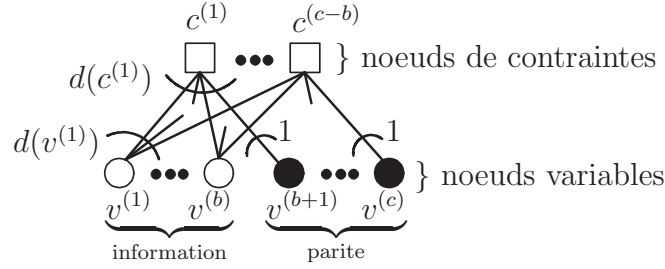


FIGURE 3.8 Protographe *irrégulier* correspondant à un ensemble de code convolutionnels non-récursif multi-registres de taux de codage $r = b/c$.

Les auteurs de (Richardson et Urbanke, 2001a) ont montré que les graphes bipartis représentant les codes linéaires doivent être composés de noeuds variables ayant un degré supérieur où égal à deux si l'on désire s'approcher de la capacité de Shannon en utilisant un algorithme à passage de messages. Pour cette raison, nous n'étudierons aucun graphe biparti composé de noeuds variables ayant un degré égal à un. D'ailleurs, lorsqu'on représente un code convolutionnel multiregistres par sa matrice de vérification $\mathbf{H}^T(D)$, la condition sur les degrés des noeuds présentée dans (Richardson et Urbanke, 2001a) fait en sorte qu'il devient obligatoire d'exploiter la récursivité des codes convolutionnels multiregistres si l'on désire s'approcher de la capacité. Pour un code convolutionnel multiregistres tous les noeuds variables ont leur degré au moins égal à deux si chaque symbole à la sortie du codeur est injecté dans au moins deux registres à décalage différents. Cette observation sur le degré minimum des noeuds variables dans le graphe biparti des codes linéaires va dans le sens des résultats obtenus dans (Cardinal, 2001). Comme il a été présenté dans (Cardinal, 2001), les codes convolutionnels doublement orthogonaux qui utilisent plusieurs registres à décalage en parallèle et qui utilisent la récursivité sont ceux qui se sont approchés le plus près de la capacité de Shannon pour le canal AWGN.

3.3 Graphe biparti de Tanner des codes convolutionnels RCDO

Le graphe biparti de Tanner (Tanner, 1981) associé à un code linéaire est un graphe $G = (V_v \cup V_c, E)$ qui décrit la relation de dépendance temporelle entre les variables à la sortie du codeur convolutionnel récursif $v_i^{(1)}, \dots, v_i^{(c)}$ et les équations de contraintes $c_i^{(1)}, \dots, c_i^{(c-b)}$, $i = 0, 1, \dots$. Les graphes bipartis de Tanner correspondant aux codes RCDO non terminés sont dits semi-infinis, car l'on suppose que l'on débute l'encodage d'une séquence provenant de la source à partir de l'instant $i = 0$, et à partir de ce moment, nous considérons que la source n'arrête jamais de transmettre des symboles. Ce graphe biparti s'obtient à partir des équations de contraintes développées précédemment pour les codes RCDO. Ces équations sont décrites par l'équation suivante :

$$c_i^{(n)} = \sum_{m=1}^c h_{m,n} v_{i-\alpha_{m,n}}^{(m)}, \quad 1 \leq n \leq (c-b) \quad (3.35)$$

où la variable $c_i^{(n)}$ est représentée par un noeud de l'ensemble V_c dans le graphe biparti, et où chaque variable qui apparaît à la droite de l'équation (3.35) représente un noeud variable de l'ensemble V_v de G . Par exemple, la Figure 3.9 représente une partie du graphe de Tanner qui décrit la relation de dépendance entre les variables qui apparaissent dans l'équation de contrainte de l'équation (3.35). Le graphe de Tanner

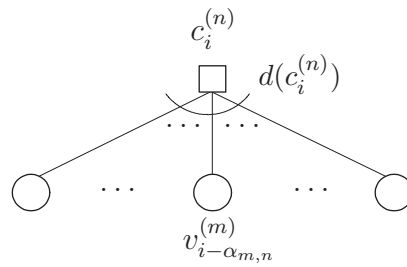


FIGURE 3.9 Sous-graphe induit par les variables apparaissant dans la n -ième équations de contraintes à l'instant i .

s'obtient donc en énumérant toutes les contraintes dans le temps, et en reliant par une arête les variables qui interviennent dans le calcul des contraintes. En effectuant cette procédure, la n -ième contrainte au temps i , $c_i^{(n)}$, possède le même degré $d(c_i^{(n)})$ que les contraintes non temporelles $c^{(n)}$ du protographe correspondant aux codes RCDO.

3.4 Arbres de calculs associés aux codes convolutifs RCDO

L'arbre de calculs associés aux codes RCDO ne représente en fait rien d'autre que le sous-graphe induit par les noeuds qui se situent dans le voisinage d'un noeud variable $v_i^{(m)}$ dans le graphe biparti de Tanner. Ce sous-graphe induit est d'un intérêt particulier, car il représente les symboles qui interviennent dans le calcul de l'estimation du symbole à la racine de l'arbre.

Nous noterons $\mathcal{N}_d(v_i^{(m)})$ le voisinage de profondeur d associé à un noeud variable $v_i^{(m)}$. Le sous-graphe induit $\mathcal{T}_{\mathcal{N}_d(v_i^{(m)})}$ par le voisinage $\mathcal{N}_d(v_i^{(m)})$ représente toutes les chaînes du graphe biparti de Tanner de longueurs d , qui débutent au noeud variable $v_i^{(m)}$. Ce sous-graphe induit est un arbre lorsqu'aucun cycle n'apparaît dans $\mathcal{T}_{\mathcal{N}_d(v_i^{(m)})}$. À partir de l'équation (3.35), les Figures 3.10 a) et 3.10 b) représentent respectivement les sous-graphes induits par les voisinages de profondeur 1, $\mathcal{N}_1(v_i^{(m)})$, et de profondeur 2, $\mathcal{N}_2(v_i^{(m)})$, du noeud $v_i^{(m)}$.

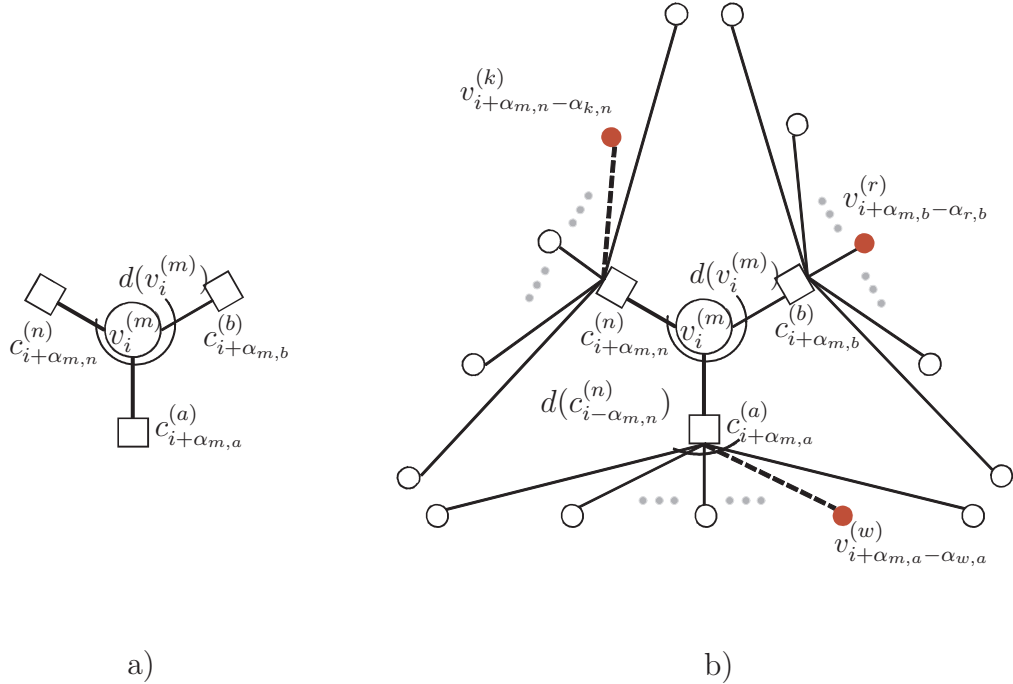


FIGURE 3.10 a) Sous-graphe induit $\mathcal{T}_{\mathcal{N}_1(v_i^{(m)})}$ b) Sous-graphe induit $\mathcal{T}_{\mathcal{N}_2(v_i^{(m)})}$.

Les feuilles de l'arbre $\mathcal{T}_{\mathcal{N}_1(v_i^{(m)})}$ de la Figure 3.10 a) représentent en fait les équations de contraintes dans lesquelles le symbole $v_i^{(m)}$ intervient. En se référant à l'équation (3.35) le symbole $v_i^{(m)}$ intervient dans un certain nombre d'équations de contraintes. Le nombre d'équations de contraintes est égal au degré $d(v_i^{(m)})$ du noeud variable qui fait office de racine. Ce nombre est aussi égal au nombre d'éléments non nuls apparaissant à la m -ième ligne de la matrice de vérification $\mathbf{H}^T(D)$ qui décrit le code RCDO. L'arbre de profondeur 2 de la Figure 3.10 b) représente le sous-graphe induit $\mathcal{T}_{\mathcal{N}_2(v_i^{(m)})}$. Étant donné que ce sous-graphe induit représente toutes les chaînes de longueur 2 qui débutent au noeud variable $v_i^{(m)}$, il s'ensuit que le sous-graphe induit $\mathcal{T}_{\mathcal{N}_2(v_i^{(m)})}$ contient le sous-graphe $\mathcal{T}_{\mathcal{N}_1(v_i^{(m)})}$. Les feuilles de $\mathcal{T}_{\mathcal{N}_2(v_i^{(m)})}$ représentent toutes les variables qui interviennent dans les équations de contraintes qui font intervenir le symbole $v_i^{(m)}$. L'arbre $\mathcal{T}_{\mathcal{N}_2(v_i^{(m)})}$ de la Figure 3.10 b) représente toute l'information qui est nécessaire à l'estimation du symbole $v_i^{(m)}$. En fait, cet arbre de calculs représente les opérations effectuées par le décodeur lors de l'estimation du symbole $v_i^{(m)}$ pour chacune des itérations. Tel que nous le présentons au prochain chapitre, pour effectuer une itération de décodage, le décodeur RCDO doit implémenter c arbres de calculs comme celui présenté à la Figure 3.10 b), où chacun des c arbres de calculs permet l'estimation de l'un des c symboles codés $v_i^{(m)}$, $m \in \{1, \dots, c\}$. Lors du décodage, les feuilles de l'arbre $\mathcal{T}_{\mathcal{N}_2(v_i^{(m)})}$ présenté à la Figure 3.10 b) transmettent des messages vers les noeuds de contraintes. Les noeuds de contraintes reçoivent sur leurs arêtes incidentes les messages provenant des feuilles et calculent les messages à transmettre vers le noeud à la racine. Ce calcul est effectué en utilisant l'équation (2.13). Le noeud à la racine de l'arbre récolte les messages provenant des noeuds de contraintes pour modifier en utilisant l'équation (2.12) la valeur de son propre message qui sera transmis ultérieurement aux autres noeuds qui se situent dans son voisinage, c'est-à-dire aux noeuds de contraintes.

Exemple 3.4— Dans cet exemple, nous montrons les sous-graphes induits de profondeur 1 et 2 au voisinage de la variable $v_i^{(1)}$ du graphe biparti de Tanner du code convolutionnel de l'exemple 3.1 qui est généré par la matrice de vérification $\mathbf{H}^T(D)$ de l'équation (3.21). Nous rappelons que

pour ce code convolutionnel les équations de contraintes s'écrivent :

$$c_i^{(1)} = v_{i-7}^{(1)} + v_{i-3}^{(2)} + v_i^{(3)} + v_i^{(4)} + v_{i-9}^{(5)} + v_{i-5}^{(6)} \quad (3.36)$$

$$c_i^{(2)} = v_{i-3}^{(1)} + v_{i-9}^{(2)} + v_i^{(3)} + v_{i-6}^{(4)} + v_i^{(5)} + v_{i-10}^{(6)} \quad (3.37)$$

$$c_i^{(3)} = v_{i-1}^{(1)} + v_i^{(2)} + v_{i-10}^{(3)} + v_{i-8}^{(4)} + v_{i-4}^{(5)} + v_i^{(6)} \quad (3.38)$$

par conséquent, les noeuds de contraintes qui représentent les feuilles du sous-graphe induit $\mathcal{T}_{\mathcal{N}_1(v_i^{(1)})}$ de la Figure 3.11 a) représentent les équations de contraintes $c_{i+7}^{(1)}$, $c_{i+3}^{(2)}$ et $c_{i+1}^{(3)}$ qui font intervenir la variable $v_i^{(1)}$.

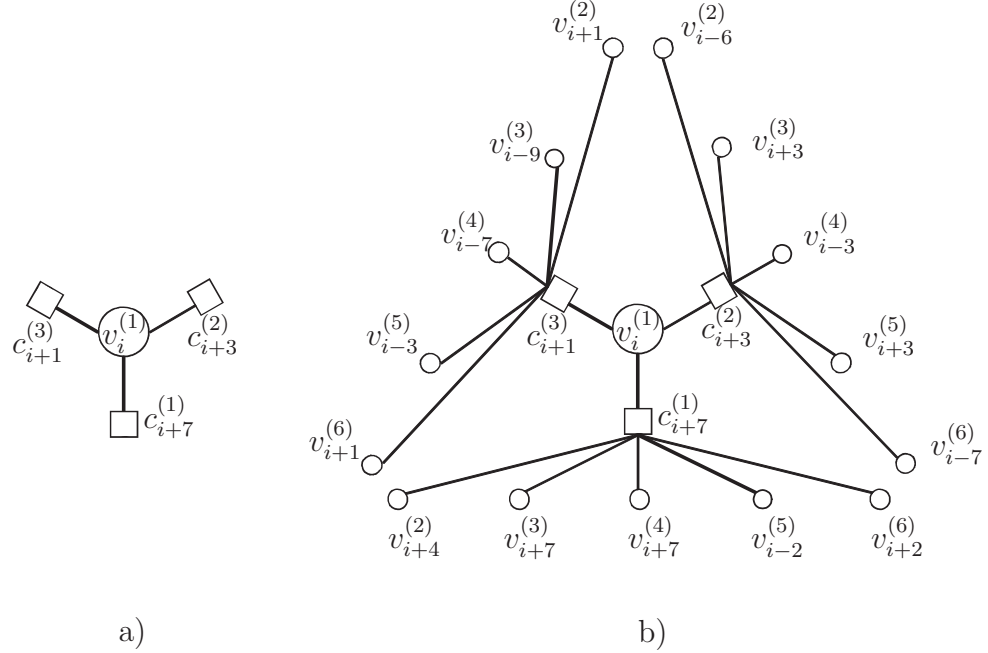


FIGURE 3.11 Sous-graphes induits $\mathcal{T}_{\mathcal{N}_1(v_i^{(1)})}$ et $\mathcal{T}_{\mathcal{N}_2(v_i^{(1)})}$ associé à l'exemple 3.4

Ces équations de contraintes sont représentées par les équations suivantes.

$$c_{i+7}^{(1)} = v_{i+7}^{(4)} + v_i^{(1)} + v_{i+4}^{(2)} + v_{i+7}^{(3)} + v_{i-2}^{(5)} + v_{i+2}^{(6)} \quad (3.39)$$

$$c_{i+3}^{(2)} = v_{i+3}^{(5)} + v_i^{(1)} + v_{i-6}^{(2)} + v_{i+3}^{(3)} + v_{i-3}^{(4)} + v_{i-7}^{(6)} \quad (3.40)$$

$$c_{i+1}^{(3)} = v_{i+1}^{(6)} + v_i^{(1)} + v_{i+1}^{(2)} + v_{i-9}^{(3)} + v_{i-7}^{(4)} + v_{i-5}^{(5)} \quad (3.41)$$

Le sous-graphe induit par les noeuds dans le voisinage de profondeur 2 de la variable $v_i^{(1)}$, qui est présenté à la Figure 3.11 b), contient toutes les

chaînes de longueur 2 débutantes au noeud $v_i^{(1)}$, c'est-à-dire que les feuilles de $\mathcal{T}_{\mathcal{N}_2(v_i^{(1)})}$ représentent les variables, autres que $v_i^{(1)}$ qui apparaissent dans les équations (3.39), (3.40) et (3.41).

3.5 Conditions d'orthogonalité

Les codes convolutionnels non-récurrents doublement-orthogonaux CDO sont décodés à l'aide d'un décodeur itératif qui est composé de \mathcal{L} décodeurs identiques en série. L'algorithme de décodage des codes CDO a été présenté dans (Cardinal, 2001) et dans (He *et al.*, 2008). En fait, le décodeur itératif des codes CDO non récurrents utilise l'algorithme à passage de message présenté au chapitre précédent. Il a été montré dans (He et Haccoun, 2005) que les messages échangés entre les noeuds variables et les noeuds de contraintes dans le graphe biparti de Tanner des codes CDO sont indépendants les uns des autres sur deux itérations successives. Il a été aussi montré dans (He et Haccoun, 2005) que le graphe biparti de Tanner des codes CDO ne contient aucun cycle de longueur 4, 6 et 8 lorsque les conditions de la double orthogonalité sont respectées. Par conséquent, le choix des connexions $\alpha_{m,n}$ joue un rôle prépondérant dans la répartition des cycles dans le graphe biparti des codes convolutionnels.

Le but de cette présente section est d'établir l'extension des conditions de la double orthogonalité, mais pour des codes convolutionnels récurrents. Ces conditions adaptées aux codes convolutionnels récurrents permettent d'éliminer les cycles de longueur 4, 6 et 8 dans la structure du graphe de Tanner de ces codes. Dans cette section, nous définissons les conditions de la simple orthogonalité et de la double orthogonalité que nous pouvons imposer aux positions des connexions des codeurs convolutionnels multi registres.

Définition 3.5 : Un codeur convolutionnel récurrent multiregistres LDPC de taux de codage $r = b/c$ est *simplement orthogonal* si la position des connexions du codeur est telle que : les différences $(\alpha_{k,n} - \alpha_{k,m})$ sont distinctes des différences $(\alpha_{s,n} - \alpha_{s,m})$, $k \neq s$, $m \neq n$, $k, s \in \{1, \dots, c\}$ et $m, n \in \{1, \dots, (c - b)\}$ et $\alpha_{k,m} \in \mathbb{N}$.

Théorème 3.1 : Tous les codes convolutionnels récurrents LDPC *simple-*

ment orthogonaux possèdent un graphe biparti dont la longueur du plus petit cycle est supérieure à 4.

Preuve : La preuve du Théorème 3.1 se retrouve à l'Annexe A.

Le Théorème 3.1 indique que pour une itération donnée, un code convolutionnel récursif multiregistres, qui possède une matrice de contrôle $\mathbf{H}^T(D)$ dont les éléments $\alpha_{m,n}$ rencontrent la Définition 3.5, permet le décodage des symboles à l'aide d'un ensemble d'équations de contraintes qui sont constituées de variables différentes. C'est-à-dire que les messages récoltés pour effectuer l'estimation d'un symbole sont indépendants les uns des autres. Ceci s'avère pratique pour ne pas propager les erreurs lors de l'estimation des symboles. Cependant, un problème réside si nous utilisons un décodeur itératif pour décoder ces codes, l'indépendance entre les messages échangés d'une itération à l'autre n'est pas conservée. Dans le but de pallier à ce problème, nous définissons les conditions associées aux codes convolutionnels récursifs multiregistres LDPC que nous nommons les codes RCDO. Avec l'ajout de deux nouvelles conditions, il devient possible d'échanger des messages indépendants sur deux itérations successives.

Définition 3.6 : Un code convolutionnel récursif multiregistres LDPC de taux de codage $r = b/c$ est *doublement orthogonal* et donc RCDO, si la position des connexions du codeur est telle que :

1. Les différences $(\alpha_{k,n} - \alpha_{k,m})$ sont distinctes des différences $(\alpha_{s,n} - \alpha_{s,m})$, $k \neq s$, $m \neq n$.
2. Les différences $(\alpha_{k,l} - \alpha_{k,q})$ sont distinctes des différences de différences $(\alpha_{s,l} - \alpha_{s,h}) - (\alpha_{r,h} - \alpha_{r,q})$, $k \neq r$, $k \neq s$, $l \neq q$, $h \neq q$, $s \neq r$, $h \neq l$.
3. Les différences de différences $(\alpha_{k,n} - \alpha_{k,q}) - (\alpha_{m,n} - \alpha_{m,p})$ sont distinctes des différences de différences $(\alpha_{f,p} - \alpha_{f,h}) - (\alpha_{r,q} - \alpha_{r,h})$, $k \neq m$, $k \neq r$, $f \neq m$, $f \neq r$, $h \neq p$, $h \neq q$, $p \neq n$, $q \neq n$.

où $k, p, r, s, t \in \{1, 2, \dots, c\}$ and $f, g, l, m, n \in \{1, 2, \dots, (c - b)\}$ et $\alpha_{k,n} \in \mathbb{N}$.

Théorème 3.2 : Tous les codes convolutionnels récursifs LDPC doublement orthogonaux possèdent un graphe biparti dont la longueur du plus petit cycle est supérieure à 8.

Preuve : La preuve du théorème se retrouve à l'Annexe A.

La répercussion du Théorème 3.2 est qu'il est possible, en ajoutant 2 nouvelles conditions par rapport à la définition des codes simplement orthogonaux, d'effectuer un décodage tel que les messages échangés pour estimer une variable sont indépendants sur 2 itérations successives.

3.6 Conclusion

Dans ce chapitre, nous avons défini la matrice de contrôle $\mathbf{H}^T(D)$ qui est associée aux codes convolutionnels récurrents RCDO. Ces codes convolutionnels sont générés à partir de simples codeurs convolutionnels récurrents ayant plusieurs registres à décalage en parallèle, et dont la position des connexions entre les symboles à la sortie du codeur et les différents éléments de délai qui composent les registres à décalage ne varient pas dans le temps. À partir de cette matrice de contrôle, nous avons associé un graphe biparti appelé le protographe $G_{\mathcal{P}}$ qui est défini par une paire de distributions (λ, ρ) ainsi que par son nombre d'arêtes. Ce petit graphe biparti permet la génération du graphe biparti de Tanner une structure encore beaucoup plus grande que celle du protographe, mais qui possède exactement la même paire de distributions (λ, ρ) qui définit le protographe à partir duquel le graphe de Tanner est généré. Les conditions d'orthogonalité que nous avons défini ne représentent en fait rien de moins que la permutation des arêtes entre les différentes copies du protographe. Ceci permet comme nous l'avons vu d'éliminer les petits cycles dans la structure du graphe biparti des codes RCDO, ce qui tend à minimiser la propagation des erreurs lors du décodage des codes convolutionnels.

Au chapitre suivant, nous présentons l'étude de la complexité liée à l'encodage et au décodage des codes RCDO. Comme nous le montrons, les différents paramètres liés à la complexité dépendent des protographes associées aux codes RCDO.

Chapitre 4

Analyse de la complexité associée au codage et au décodage des codes RCDO

Nous avons établi au chapitre précédent les caractéristiques associées aux graphes bipartis des codes convolutionnels récurrents RCDO. Comme nous l'avons vu, les codes RCDO possèdent un protographe qui représente le module de départ permettant de générer un graphe biparti de Tanner. Les graphes bipartis de Tanner des codes RCDO possèdent deux caractéristiques fondamentales. La première caractéristique est qu'ils ne possèdent aucun cycle de longueur inférieure à 10. La seconde caractéristique est qu'ils possèdent une paire de distributions (λ, ρ) identique à celles des protographes à partir desquels ils sont générés.

Comme nous l'avons présenté au chapitre précédent, les codes RCDO forment une famille de codes construits à partir de leur matrice de contrôle qui est creuse. Tout comme pour les codes en blocs LDPC, il n'est pas possible d'utiliser un décodeur qui maximise le rapport de vraisemblance pour décoder les codes RCDO. Pour les codes convolutionnels ayant une faible longueur de contrainte, l'algorithme de Viterbi permet d'effectuer ce type de décodage qui minimise la probabilité d'erreur sur la séquence transmise. Cependant, la complexité de décodage est de l'ordre $O(\frac{1}{r}2^K)$, où K représente la mémoire totale du codeur convolutionnel et où $r = b/c$ représente le taux de codage du code convolutionnel (Richardson et Urbanke, 2008). Il s'avère que les codes convolutionnels RCDO étudiés possèdent une longueur de contrainte qui est généralement beaucoup trop élevée pour envisager ce type de décodage.

Pour effectuer le décodage des codes RCDO, nous devons nous rabattre sur un algorithme itératif qui travaille localement dans le graphe biparti du code. L'objectif de l'algorithme itératif à passage de messages est d'inférer la valeur d'un symbole à la sortie du canal de communication à partir de l'information provenant des sym-

boles qui se situent dans le voisinage du symbole à estimer dans le graphe biparti du code RCDO. Ce type d'algorithme itératif permet d'obtenir les mêmes performances d'erreur que l'algorithme à maximum de vraisemblance lorsque le graphe biparti de Tanner d'un code est sans cycle (Richardson et Urbanke, 2008). Toutefois, même si nous générons un graphe biparti n'ayant pas de cycle, il est facilement démontrable que ce type de code génère des mots de code ayant une distance de Hamming de faible poids. Cette propriété n'est pas propice à de bonnes performances d'erreur, car les mots de codes ayant une distance de Hamming de faible poids entraînent une dégradation des performances d'erreur. Dans le but d'accroître la distance de Hamming des mots de code, le graphe biparti de Tanner des codes doit contenir des cycles (Richardson et Urbanke, 2008). Les cycles qui composent alors les graphes bipartis rendent l'algorithme itératif à passage de messages sous-optimaux par rapport à l'algorithme qui maximise le rapport de vraisemblance (Richardson et Urbanke, 2008). L'objectif de ce chapitre est d'analyser les complexités d'encodage et de décodage des codes RCDO.

4.1 Complexités relatives à l'encodage des codes RCDO

Dans cette section, nous montrons que la complexité associée aux calculs des $(c-b)$ symboles de parité des codes RCDO est proportionnelle à $|E_{\mathcal{P}}|$, où $|E_{\mathcal{P}}|$ représente le nombre d'arêtes constituant le protographe du code convolutionnel de taux de codage $r = b/c$.

4.1.1 Complexité de calculs liés à l'encodage des codes RCDO

Pour montrer que la complexité de calculs est une fonction linéaire du nombre d'arêtes constituant le protographe, il suffit uniquement d'observer qu'à chaque instant i , les $(c-b)$ symboles de parité générés à la sortie d'un codeur convolutionnel de taux de codage $r = b/c$ sont calculés à partir de l'équation suivante.

$$v_i^{(b+n)} = \sum_{\substack{m=1 \\ m \neq b+n}}^{(c)} v_{i-\alpha_{m,n}}^{(m)}, \quad 1 \leq n \leq c-b \quad (4.1)$$

Comme nous l'avons montré plus haut, nous pouvons associer une équation de contraintes à chacun des symboles de parité générés à l'instant i :

$$c_i^{(n)} = v_i^{(b+n)} + \sum_{\substack{m=1 \\ m \neq b+n}}^{(c)} v_{i-\alpha_{m,n}}^{(m)}, \quad 1 \leq n \leq c-b \quad (4.2)$$

Le graphe biparti de la figure 4.1 représente le sous-graphe induit par les noeuds variables et le noeud de contraintes qui apparaissent dans l'équation (4.2). Le graphe biparti de cette figure est dirigé, c'est-à-dire que les arêtes du graphe possèdent une orientation. Pour déterminer la valeur du symbole de parité $v_i^{(b+n)}$ il suffit de transmettre vers le noeud de contrainte les messages binaires provenant des variables qui apparaissent dans la somme de l'équation (4.1). Le noeud de contrainte de degré $d(c_i^{(b+n)})$ effectue la somme modulo 2 des $(d(c_i^{(b+n)}) - 1)$ messages binaires déjà connus et transmet le résultat vers le noeud variable $v_i^{(b+n)}$ qui représente le symbole de parité à la $(b+n)$ -ième sortie du codeur convolutionnel.

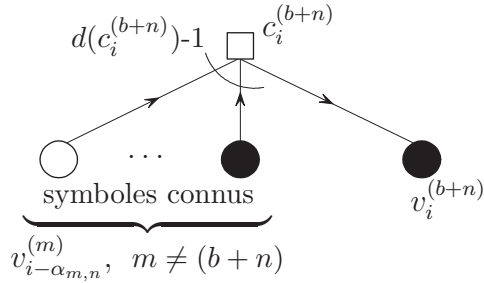


FIGURE 4.1 Procédure d'encodage des symboles de parité au temps i .

Par conséquent, l'ensemble des $(c-b)$ symboles de parité est obtenu en effectuant au total N_{Σ}^C additions modulo 2. Cette valeur est égale à :

$$N_{\Sigma}^C = \sum_{n=1}^{(c-b)} \left((d(c_i^{(n)}) - 1) - 1 \right) \quad (4.3)$$

$$= \sum_{n=1}^{(c-b)} d(c_i^{(n)}) - \sum_{n=1}^{(c-b)} 2 \quad (4.4)$$

$$= |E_{\mathcal{P}}| - 2(c-b) \quad (4.5)$$

ce qui nous permet de conclure que la complexité de calculs liée à l'encodage des codes

convolutionnels RCDO est de l'ordre $O(|E_{\mathcal{P}}|)$ additions modulo 2. Pour un code en blocs linéaire LDPC de taille n , la complexité associée à l'encodage dépend de plusieurs facteurs. Généralement, si le code en blocs LDPC est généré de façon aléatoire alors la complexité pour l'encodage d'un bloc peut atteindre $O(n^2)$ (Richardson et Urbanke, 2001b) opérations, cependant dans certains cas cette complexité peut être réduite à $O(n^{\frac{3}{2}})$ opérations en utilisant des techniques spécifiques de construction comme celles présentées dans (Richardson et Urbanke, 2001b).

4.1.2 Mémoire requise par un codeur convolutionnel RCDO

La mémoire requise par un codeur RCDO correspond au nombre d'éléments de délai requis par chacun des $(c - b)$ registres à décalage nécessaires à la conception du codeur convolutionnel RCDO de taux de codage $r = b/c$. Tous les éléments de délai qui constituent les registres à décalage doivent contenir un bit. Par conséquent, pour les codes RCDO définis par la matrice de contrôle $\mathbf{H}^T(D)$, le nombre de bits que nous devons enregistrer correspond à la somme des éléments de délai qui constituent chacun des $(c - b)$ registre à décalage. En définissant δ_n comme étant le nombre d'éléments de délai qui composent le n -ième registre à décalage, nous pouvons écrire :

$$\delta_n = \max_{m \in \{1, \dots, c\}} (\alpha_{m,n}) \quad (4.6)$$

et la mémoire totale nécessaire m_{tot} est donc la somme des délais associés à chacun des registres à décalage :

$$m_{\text{tot}} = \sum_{n=1}^{(c-b)} \delta_n \quad (4.7)$$

Si nous notons par m_s la valeur du plus grand registre à décalage, c'est-à-dire :

$$m_s = \max_{n \in \{1, \dots, (c-b)\}} (\delta_n) \quad (4.8)$$

alors une borne supérieure sur la mémoire totale s'écrit :

$$m_{\text{tot}} < (c - b)m_s \quad (4.9)$$

L'équation (4.9) nous indique que la mémoire totale nécessaire à la réalisation d'un codeur convolutionnel RCDO de taux de codage $r = b/c$ repose essentiellement sur

le nombre $(c - b)$ de registres à décalage en parallèle, et sur m_s la mémoire du plus grand de ces registres. En comparaison, un code en blocs LDPC de taille n nécessite n bits de mémoire pour effectuer l'encodage d'un bloc. Par conséquent, un codeur convolutionnel RCDO offre l'avantage d'utiliser moins de mémoire lorsque la valeur m_{tot} est inférieure à la taille n d'un code en blocs LDPC.

4.2 Complexités associées au décodage itératif des codes convolutionnels RCDO

Dans cette section, nous analysons la complexité des calculs et la latence induite par les codes convolutionnels RCDO. Nous montrons que la complexité des calculs associée au décodage itératif dépend du nombre d'itérations effectuées lors du décodage et du protographe associé au code convolutionnel utilisé. La latence introduite pour le décodage des codes RCDO dépend du nombre d'itérations et de la mémoire m_s du plus grand registre à décalage.

Le décodage itératif des codes RCDO est effectué par une chaîne de décodeurs identiques où le nombre de décodeurs élémentaires correspond au nombre d'itérations effectuées. Ce type de décodeur itératif se distingue du décodeur généralement associé aux codes en blocs LDPC, qui utilisent qu'un seul décodeur pour effectuer le décodage itératif des mots de code. Le décodeur itératif RCDO présente l'avantage d'être plus simple à réaliser qu'un décodeur pour les codes LDPC, car seulement un simple décodeur élémentaire doit être conçu pour réaliser l'ensemble du décodeur.

À la Figure 4.2, nous présentons le schéma fonctionnel du décodeur itératif envisagé pour effectuer le décodage des codes RCDO. Le décodeur itératif présenté fonctionne de la manière suivante. À chaque unité de temps i , le récepteur fournit à la sortie de filtres adaptés aux signaux transmis, un vecteur de c valeurs réelles qui correspondent aux messages originaux qui sont utilisés par le décodeur itératif. Nous noterons le message initial correspondant à la m -ième sortie $v_i^{(m)}$ du codeur convolutionnel RCDO par $m_{v_i^{(m)}}^{(0)}$, $m \in \{1, \dots, c\}$. Ce message correspond en fait à une mesure initiale de la fiabilité sur le symbole $v_i^{(m)}$ après qu'il est été altéré par le canal de communication considéré. La valeur $m_{v_i^{(m)}}^{(0)}$ représente le logarithme du rapport de vraisemblance (LRV) du symbole $v_i^{(m)}$ transmis. Cette mesure dépend essentiellement du canal de communication et est décrite dans le prochain chapitre.

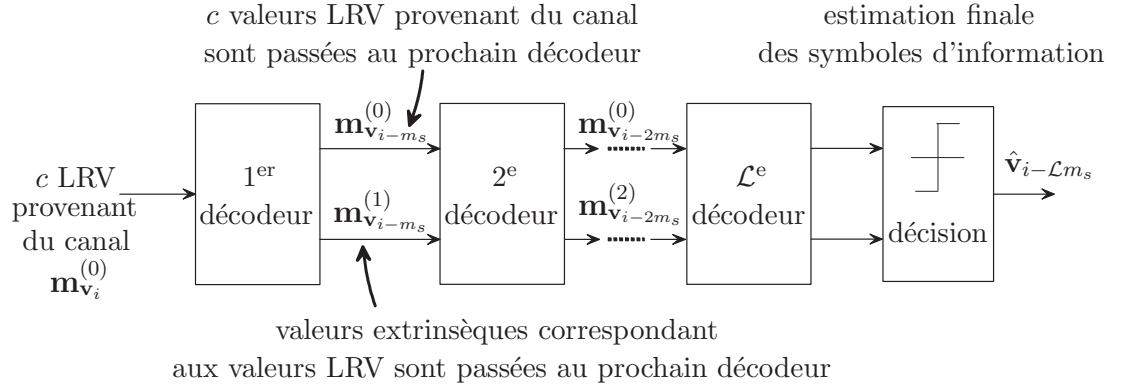


FIGURE 4.2 Schéma bloc représentant le décodeur itératif sur \mathcal{L} itérations des codes RCDO.

Le premier décodeur enregistre dans chacun des c registres à décalage en parallèle la composante du vecteur de messages initiaux $\mathbf{m}_{\mathbf{v}_i}^{(0)}$, $\mathbf{m}_{\mathbf{v}_i}^{(0)} = (m_{v_i^{(1)}}^{(0)}, \dots, m_{v_i^{(c)}}^{(0)})$. Ces c valeurs correspondent aux c composantes du vecteur \mathbf{v}_i qui forme une partie du mot de code transmis. À la réception du vecteur $\mathbf{m}_{\mathbf{v}_i}^{(0)}$, chaque élément composant chacun des registres à décalage du décodeur est décalé d'un élément de délai. Le dernier vecteur de message qui constitue le contenu des c registres à décalage en parallèle est transmis au prochain décodeur dans la cascade, qui accepte ce vecteur et décale à son tour l'ensemble des messages initiaux stockés dans les c registres à décalage. Ce processus déclenche une certaine réaction en chaîne et fait en sorte que tous les messages initiaux sont décalés dans l'ensemble des c registres à décalage des \mathcal{L} décodeurs élémentaires considérés.

En même temps qu'un vecteur de messages initiaux quitte l'un des \mathcal{L} étages du décodeur itératif, les valeurs extrinsèques qui correspondent aux vecteurs de messages originaux $\mathbf{m}_{\mathbf{v}_{i-\ell m_s}}^{(0)}$, $\ell \in \{1, \dots, \mathcal{L}\}$, sont transmises au prochain décodeur. Ces valeurs extrinsèques représentent la mesure de fiabilité qui provient des variables qui se situent dans le voisinage de profondeur 2, $\mathcal{N}_2(v_{i-\ell m_s}^{(m)})$, dans le graphe biparti des codes RCDO. À la dernière itération, en utilisant les valeurs initiales provenant du canal $\mathbf{m}_{\mathbf{v}_{i-Lm_s}}^{(0)}$, ainsi que les valeurs extrinsèques des variables $\mathbf{m}_{\mathbf{v}_{i-Lm_s}}^{(\mathcal{L})}$ une décision finale $\hat{\mathbf{v}}_{i-Lm_s}$ est effectuée sur la partie du mot de code correspondante au vecteur \mathbf{v}_{i-Lm_s} transmis par

le codeur convolutionnel RCDO.

4.2.1 Complexité de calculs de l'information extrinsèque

L'information extrinsèque associée à un noeud variable se calcule en deux étapes comme nous l'avons présenté à la Section 2.5 du Chapitre 2. En se référant à la Figure 4.3, la première étape consiste à transmettre les messages associés aux feuilles du sous-graphe induit $\mathcal{T}_{\mathcal{N}_2(v_t^{(m)})}$ vers les noeuds de contraintes. Lors de la réception

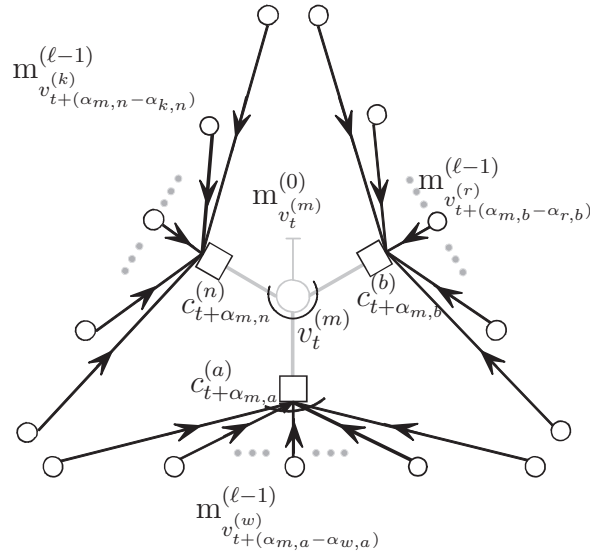


FIGURE 4.3 Sous-graphe induit $\mathcal{T}_{\mathcal{N}_2(v_t^{(m)})}$ de profondeur 2 au voisinage du noeud variable $v_t^{(m)}$, $t = i - \ell m_s$.

de ces messages les $d(v_{i-\ell m_s}^{(m)})$ noeuds de contraintes calculent selon un opérateur que nous appelons \mathcal{R} les messages à transmettre vers le noeud à la racine de l'arbre. Cette étape est représentée à la Figure 4.4, où on présente les messages transmis entre les noeuds de contraintes $c_{t+\alpha_{m,n}}^{(n)}$ et le noeud variable $v_t^{(m)}$, avec $t = i - \ell m_s$. Nous notons ce message $m_{c_{t+\alpha_{m,n}}^{(n)}, v_t^{(m)}}^{(\ell-1)}$ et il est obtenu par la fonction $\mathcal{R}_{d(c_{t+\alpha_{m,n}}^{(n)})-1}()$ qui utilise $(d(c_{t+\alpha_{m,n}}^{(n)}) - 1)$ messages provenant des feuilles de l'arbre. Cette procédure peut être écrite selon l'équation générale suivante :

$$m_{c_{t+\alpha_{m,n}}^{(n)}, v_t^{(m)}}^{(\ell)} = \mathcal{R}_{d(c_{t+\alpha_{m,n}}^{(n)})-1} \left(m_{v_{t+\alpha_{m,n}-\alpha_{k,n}}, c_{t+\alpha_{m,n}}^{(n)}}^{(\ell-1)} : k \neq m \right) \quad (4.10)$$

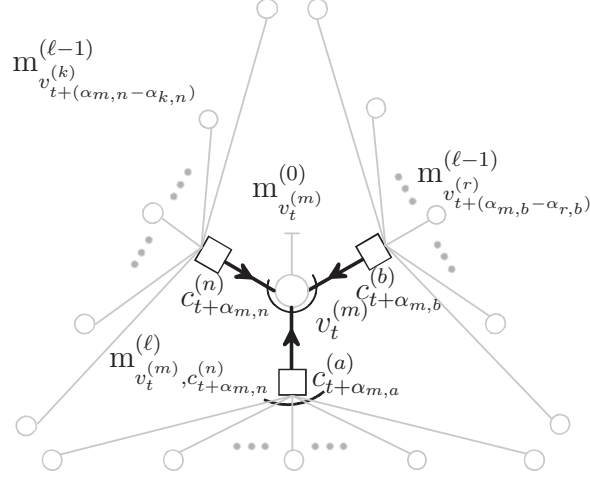


FIGURE 4.4 Sous-graphe induit $\mathcal{T}_{\mathcal{N}_2(v_t^{(m)})}$ associé à un code RCDO.

où $m_{v_{t+\alpha_m,n-\alpha_{k,n}}^{(k)}, c_{t+\alpha_m,n}^{(n)}}^{(\ell-1)}$ représente l'information extrinsèque, calculée à l'itération $(\ell - 1)$, et qui est transmise entre la feuille de l'arbre $v_{t+\alpha_m,n-\alpha_{k,n}}^{(k)}$ et le noeud de contrainte $c_{t+\alpha_m,n}^{(n)}$. Le nombre de messages reçus par la racine provenant des noeuds de contraintes est donc égal au degré $d(v_t^{(m)})$ du noeud et par conséquent il faut $d(v_t^{(m)})$ opérateurs \mathcal{R} pour obtenir ces messages. Il s'ensuit que le nombre de messages à calculer pour les c noeuds variables $v_t^{(m)}$, $m \in \{1, \dots, c\}$, nécessite un nombre d'opérateurs \mathcal{R} égal à la somme de tous les degrés des c noeuds variables $v_t^{(m)}$ à l'instant t . Ce nombre représenté par $N_{\mathcal{R}}$ est donc égal à :

$$\begin{aligned} N_{\mathcal{R}} &= \sum_{m=1}^c d(v_t^{(m)}) \\ &= |E_{\mathcal{P}}| \end{aligned} \tag{4.11}$$

Le nombre d'opérateurs \mathcal{R} nécessaires pour une itération de décodage est donc égal au nombre d'arêtes qui composent le protographe du code RCDO. Généralement, l'opérateur \mathcal{R} représente soit la règle somme-produit, défini par l'équation (2.13), soit la règle min-somme (Kschischang *et al.*, 2001). La seconde règle représente une approximation de la règle somme-produit (Hagenauer *et al.*, 1996). L'analyse de la complexité que nous effectuons dans cette thèse est générale et ne tient pas compte du type de règle utilisée pour faire la mise à jour de ces messages. Nous référons

le lecteur aux travaux de (Nemr *et al.*, 2008) pour connaître la complexité associée à la réalisation matérielle de la règle min-somme dans le cas des codes doublement orthogonaux. Le but de notre analyse est de montrer que peu importe la règle utilisée, le nombre d'opérateurs nécessaires pour une itération dépend uniquement du protographe du code RCDO utilisé.

La seconde étape nécessaire au calcul de l'information extrinsèque associée à la m -ième variables $v_t^{(m)}$ consiste à effectuer la somme des messages reçus $m_{c_{t+\alpha_m,n}, v_t^{(m)}}^{(\ell-1)}$ provenant des noeuds de contraintes dans le voisinage du noeud variable $v_t^{(m)}$. La Figure 4.5 montre comment s'effectue le calcul du message qui exprime l'information extrinsèque échangée entre le noeud variable $v_t^{(m)}$ et le noeud de contrainte $c_{t+\alpha_m,n}^{(m)}$ à l'itération ℓ . Ce message est représenté par $m_{v_t^{(m)}, c_{t+\alpha_m,n}^{(m)}}^{(\ell)}$ et s'obtient selon la relation suivante (Richardson et Urbanke, 2008) :

$$m_{v_t^{(m)}, c_{t+\alpha_m,n}^{(m)}}^{(\ell)} = m_{v_t^{(m)}}^{(0)} + \sum_{\substack{s=1 \\ s \neq n}}^{(c-b)} m_{c_{t+\alpha_m,s}^{(m)}, v_t^{(m)}}^{(\ell-1)} \quad (4.12)$$

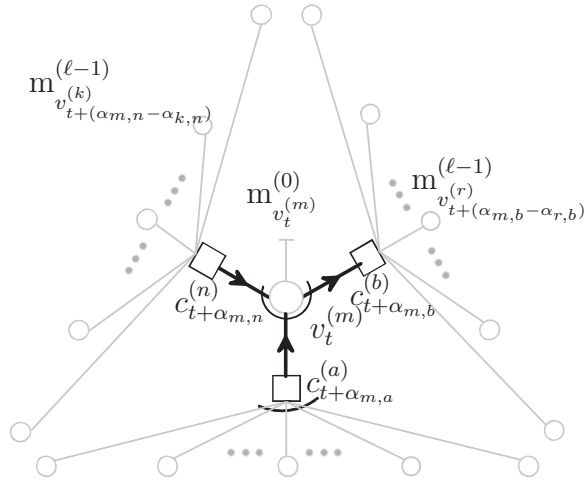


FIGURE 4.5 Sous-graphe induit $\mathcal{T}_{\mathcal{N}_2(v_t^{(m)})}$ correspondant à l'arbre de décodage du symbole $v_t^{(m)}$ d'un code RCDO.

Nous pouvons facilement observer que pour chaque variable $v_t^{(m)}$ on doit effectuer une mise à jour de $d(v_t^{(m)})$ messages extrinsèques, c'est-à-dire une mise à jour

pour chaque arête incidente au noeud variable. Chacun des $d(v_i^{(m)})$ messages extrinsèques nécessite, selon l'équation (4.12), $(d(v_i^{(m)}) - 1)$ additions de nombres réels. Par conséquent, pour une itération donnée, à chaque instant il y a c variables pour lesquelles nous devons effectuer cette mise à jour de l'information extrinsèque. Pour faire la mise à jour de tous les messages extrinsèques pour l'ensemble des c variables $v_i^{(m)}$, il faut effectuer un total de N_Σ^d additions de nombres réels. Ce nombre est égal à :

$$N_\Sigma^d = \sum_{m=1}^c d(v_i^{(m)})(d(v_i^{(m)}) - 1) \quad (4.13)$$

$$= \sum_{m=1}^c d^2(v^{(m)}) - |E_{\mathcal{P}}| \quad (4.14)$$

et en utilisant l'équation (3.24) nous obtenons,

$$\sum_{m=1}^c d^2(v^{(m)}) = \sum_{j=2}^{d_\lambda} j^2 n_j^l \quad (4.15)$$

$$= |E_{\mathcal{P}}|(\lambda'(1) + 1) \quad (4.16)$$

en substituant l'équation (4.16) dans (4.14) nous obtenons :

$$N_\Sigma^d = |E_{\mathcal{P}}|\lambda'(1) \quad (4.17)$$

En résumé, pour effectuer la mise à jour de l'information extrinsèques des c variables $v_i^{(m)}$ à une itération donnée, il faut utiliser $|E_{\mathcal{P}}|$ opérateurs \mathcal{R} et effectuer $|E_{\mathcal{P}}|\lambda'(1)$ additions de nombres réels. Nous mentionnons qu'il est possible d'améliorer le comportement du décodeur itératif lorsque nous pondérons l'équation (4.12) à l'aide d'un facteur de pondération w , $0 < w \leq 1$. Cette technique a été utilisée dans (Cardinal, 2001) et dans grand nombre de travaux. De sorte, que le calcul de l'information extrinsèque obtenu par l'équation (4.12) devient :

$$m_{v_i^{(m)}, c_{i+\alpha_{m,n}}^{(n)}}^{(\ell)} = w \left(m_{v_i^{(m)}}^{(0)} + \sum_{\substack{s=1 \\ s \neq m}}^{(c-b)} m_{c_{i+\alpha_{m,s}}, v_i^{(m)}}^{(\ell-1)} \right) \quad (4.18)$$

En utilisant cette méthode, pour chacune des itérations, nous devons ajouter $|E_{\mathcal{P}}|$

pondérateurs, car nous devons pondérer les $|E_{\mathcal{P}}|$ messages extrinsèques associés aux c noeuds variables $v_t^{(m)}$. Avec cette méthode, la détermination du coefficient w devient primordial dans le but de minimiser la probabilité d'erreur à une itération donnée. Dans (Cardinal, 2001) ce coefficient a été évalué à l'aide de réseaux neuronaux sans toutefois donner des résultats qui concordent avec les résultats pratiques. Depuis ces travaux, des chercheurs se sont attardés au problème associé à l'évaluation du paramètre w . Dans (Chen et Fossorier, 2002) et (Kim et Park, 2004) on retrouve deux méthodes qui permettent d'évaluer ce paramètre. Nous mentionnons que les résultats obtenus avec les codes RCDO semblent concorder avec les résultats proposés dans (Kim et Park, 2004) pour les codes en blocs LDPC. Nous mentionnons que la valeur pratique $w = 0.82$ a été utilisée pour générer tous les résultats de cette thèse.

4.2.2 Délai de décodage

Un paramètre important lors du choix d'un code correcteur d'erreur est le délai toléré pour pouvoir avoir une estimation sur les symboles d'information transmis par la source. Pour le décodeur présenté précédemment, le délai de décodage est proportionnel au nombre d'itérations effectuées ainsi qu'au nombre de symboles que nous devons enregistrer pour pouvoir calculer les valeurs extrinsèques des symboles décodés. Le nombre de symboles que l'on doit accumuler pour calculer l'information extrinsèque à une itération donnée est de $c(m_s + 1)$ symboles. Au total si nous utilisons \mathcal{L} décodeurs, nous devrions accumuler $\mathcal{L}c(m_s + 1)$ symboles. Si tous les calculs peuvent être effectués en parallèle pour chacun des décodeurs alors on doit attendre $\mathcal{L}(m_s + 1)$ unités de temps avant de pouvoir prendre une décision finale sur le vecteur de messages provenant du canal $\mathbf{m}_{\mathbf{v}_i - \mathcal{L}m_s}^{(0)}$.

4.2.3 Mémoire nécessaire à la réalisation d'une itération du décodeur itératif

Pour calculer l'information extrinsèque des symboles à la sortie d'un étage du décodeur itératif, nous devons enregistrer un certain nombre de valeurs. En nous référant à l'équation (4.18), nous pouvons remarquer que nous devons enregistrer en mémoire les messages provenant du canal $\mathbf{m}_{\mathbf{v}_i - \mathcal{L}m_s}^{(0)}$, car le calcul de l'information extrinsèque par rapport à une variable donnée nécessite les messages initiaux provenant du canal. Lorsque nous évaluons l'information extrinsèque par rapport au symbole

$v_{i-\ell m_s}^{(m)}$, on observe qu'il faut avoir accumulé $c\ell(m_s + 1)$ messages provenant du canal. Ces valeurs sont enregistrées dans c registres à décalage en parallèle. Pour chacune des itérations, chacun des c registres à décalage doit pouvoir contenir $(m_s + 1)$ messages. C'est-à-dire que pour une itération donnée on doit donc avoir $c(m_s + 1)$ unités de mémoire pour stocker ces nombres réels. De plus, comme nous l'avons montré, pour chaque variable $v_{i-\ell m_s}^{(m)}$ estimée nous devons stocker en mémoire un message extrinsèque par arête incidente au noeud $v_{i-\ell m_s}^{(m)}$, c'est-à-dire que l'on doit enregistrer $d(v_{i-\ell m_s}^{(m)})$ valeurs extrinsèques par symbole décodé. Comme nous décodons c symboles simultanément et que le nombre d'éléments de mémoires nécessaires à l'évaluation des messages extrinsèque mis à jour dépend du degré de chacun des noeuds variables, le nombre total de mémoires nécessaires pour stocker les valeurs extrinsèques retournées par la racine vers les noeuds de contraintes est définie par :

$$m_{\text{ext}} = \sum_{m=1}^c d(v_{i-\ell m_s}^{(m)}) \quad (4.19)$$

$$= |E_{\mathcal{P}}| \quad (4.20)$$

il s'ensuit que pour chaque vecteur initial de c composantes réelles $\mathbf{m}_{\mathbf{v}_{i-\ell m_s}^{(0)}}$, nous devons conserver $|E_{\mathcal{P}}|$ messages extrinsèques. Par conséquent, la mémoire totale m^{Tot} pour une itération de décodage peut s'exprimer par l'équation (4.21).

$$m^{\text{Tot}} = (|E_{\mathcal{P}}| + c)(m_s + 1) \quad (4.21)$$

Si l'on effectue \mathcal{L} itérations lors du décodage alors l'ensemble des décodeurs utilisés nécessite un total de $\mathcal{L}(|E_{\mathcal{P}}| + c)(m_s + 1)$ unités de mémoires.

4.3 Conclusion

Suite à l'analyse présentée dans ce chapitre, nous pouvons revoir le décodeur itératif dans son ensemble comme étant une structure composée de $(|E_{\mathcal{P}}| + c)$ registres à décalage en parallèle chacun pouvant contenir $\mathcal{L}(m_s + 1)$ nombres réels où, \mathcal{L} processeurs indépendants effectuent la mise à jour simultanément de messages extrinsèques. Pour faire la mise à jour de ces messages, chaque processeur doit lire certaines valeurs dans les $(|E_{\mathcal{P}}| + c)$ registres à décalage qui constituent le décodeur. Les valeurs lues par le ℓ -ième processeur permettent le calcul des mises à jour de l'information ex-

trinsèque associées aux c symboles $v_{i-\ell m_s}^{(m)}$, $m \in \{1, \dots, c\}$, $\ell \in \{1, \dots, \mathcal{L}\}$. Le nombre de messages extrinsèques calculés par un processeur pour l'ensemble des c symboles $v_{i-\ell m_s}^{(m)}$ nécessite le calcul de $|E_{\mathcal{P}}|$ messages extrinsèques. L'évaluation de ces $|E_{\mathcal{P}}|$ messages extrinsèques requiert que chaque processeur doit intégrer $|E_{\mathcal{P}}|$ opérateurs \mathcal{R} ainsi que $|E_{\mathcal{P}}|$ pondérateurs en plus de pouvoir effectuer $|E_{\mathcal{P}}|\lambda'(1)$ sommes de nombre réels.

La dernière analyse nous permet de constater d'une part que tout ce qui a trait aux calculs lors du décodage dépend uniquement des paramètres associés aux protographes des codes RCDO. D'autre part, nous pouvons aussi remarquer que les conditions d'orthogonalité imposées aux codes RCDO qui influencent la valeur m_s se répercutent indirectement sur la mémoire totale ainsi que sur la latence totale associée au décodage des symboles d'information. Le nombre d'itérations influence le nombre de processeurs requis par le décodeur itératif ainsi que la mémoire et la latence de décodage. Notre contribution dans ce chapitre réside dans l'analyse effectuée, ce qui nous permet de rechercher des protographes qui correspondent aux codes RCDO et qui satisfont les spécifications matérielles.

Chapitre 5

Limites asymptotiques associées aux performances d’erreur des codes RCDO

Cette section de la thèse traite des limites asymptotiques, à faibles rapports signal, des performances d’erreur associées aux codes convolutionnels LDPC à temps-invariant. Plus spécifiquement, nous analysons celles associées aux codes convolutionnels récurrents doublement orthogonaux. Comme nous l’avons indiqué au chapitre 2, il est possible de déterminer une valeur du rapport signal sur bruit $(\frac{E_b}{N_0})^*$, appelée le *seuil*, à partir duquel l’algorithme de décodage itératif permet de faire converger théoriquement la probabilité d’erreur vers 0 lorsque le graphe biparti du code correcteur d’erreur ne contient pas de cycles. Cette valeur dépend essentiellement du type de canal de transmission, et de la distribution des arêtes dans le graphe biparti des codes correcteurs d’erreur.

L’objectif de la première partie du chapitre est donc de trouver des paires de distributions (λ, ρ) adaptées aux codes convolutionnels RCDO que nous avons définis au Chapitre 3. Les paires de distributions présentées sont celles qui s’approchent le plus près, à ce jour, de la capacité associée au canal binaire à effacement et au canal binaire avec bruit additif blanc et gaussien.

5.1 Seuils de convergence des codes RCDO pour le canal binaire symétrique à effacement

Le canal binaire à effacement (BEC) fut présenté en 1954 par P. Elias purement à titre d'exemple théorique d'un canal de communications. De nos jours, le canal BEC peut servir à modéliser plusieurs scénarios réels de communications. Par exemple, lorsqu'un paquet de données est transmis dans le réseau Internet, il peut être reçu correctement ou bien il peut être déclaré effacé au récepteur. Si tel est le cas, le récepteur demande une retransmission du paquet de données de la part de l'émetteur. Ce faisant, l'application qui traite les données reçoit toujours des données correctes et exemptes d'erreur. La Figure 5.1 présente le canal BEC, où la sortie E représente un effacement.

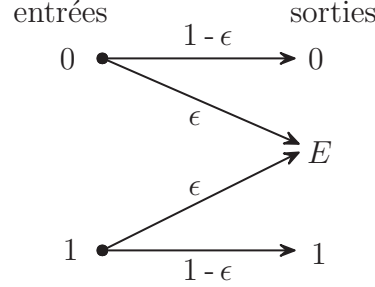


FIGURE 5.1 Canal binaire à effacement.

Dans (Richardson et Urbanke, 2001a), il est montré que pour un canal binaire à effacement ayant une probabilité d'effacement ϵ , la probabilité d'observer un effacement à l'itération ℓ peut être exprimée par la relation :

$$p_\ell = \epsilon\lambda(1 - \rho(1 - p_{\ell-1})), \quad \ell \geq 1 \quad (5.1)$$

où p_ℓ et $p_{\ell-1}$ représentent les probabilités d'effacement aux itérations ℓ et $(\ell - 1)$ respectivement, $\ell \in \{1, \dots, \mathcal{L}\}$. Et où les polynômes,

$$\lambda(x) = \sum_{j=2}^{d_\lambda} \lambda_j x^{j-1} \quad \text{et} \quad \rho(x) = \sum_{j=2}^{d_\rho} \rho_j x^{j-1} \quad (5.2)$$

représentent la paire de distributions (λ, ρ) qui décrit le graphe biparti du code convolutionnel récursif considéré, la fraction λ_j (ρ_j) représentant la proportion des arêtes incidentes aux noeuds variables (contraintes) de degrés j dans le graphe du code.

La valeur p_0 utilisée dans l'équation (5.1) représente la probabilité d'effacement initiale qui est égale à ϵ , c'est-à-dire, celle du canal à effacement. Cette équation récursive nous indique l'évolution de la probabilité d'effacement lorsque le décodage de l'information est effectué à l'aide d'un algorithme itératif à passage de messages à partir d'un graphe biparti *sans cycle* défini par une paire de distributions $\lambda(x)$ et $\rho(x)$.

La valeur du seuil ϵ^* , qui est associé à un ensemble de codes convolutionnels récursifs possédant une paire de distributions (λ, ρ) , représente donc la plus grande valeur initiale p_0 , qui permet de faire tendre la probabilité d'effacement p_ℓ vers zéro lorsque nous effectuons un nombre d'itérations qui tend vers l'infini,

$$\lim_{\ell \rightarrow \infty} p_\ell \rightarrow 0$$

La valeur de ce seuil est donc comprise dans l'intervalle $[0, \epsilon^{\text{Sh}}]$, où ϵ^{Sh} représente la capacité de Shannon pour un canal à effacement. Pour ce type de canal de communication, ϵ^{Sh} est égal à $1 - r$, où r désigne le taux de codage des codes convolutionnels considérés. Notons que la valeur ϵ^* doit donc être forcément inférieure ou égale à la limite de Shannon ϵ^{Sh} , car il n'est pas possible d'offrir un taux de transmission de l'information au-delà de cette limite tout en faisant tendre la probabilité d'effacement théorique vers zéro (Gallager, 1968). L'exemple suivant illustre le mécanisme présenté ci-dessus pour rechercher la valeur du seuil qui correspond aux codes LDPC réguliers possédant une paire de distribution (λ, ρ) .

Exemple 5.1- Pour cet exemple, nous considérons un canal BEC et nous cherchons à évaluer la valeur du seuil ϵ^* , qui est associée à l'ensemble des codes convolutionnels récursifs dont le graphe biparti de Tanner est défini par la paire de distributions (3,6). Sachant que la capacité de Shannon pour ce canal et donnée par $\epsilon^{\text{Sh}} = 1 - r$ et que le taux de codage des codes considérés est de $1/2$ on en déduit que $\epsilon^{\text{Sh}} = 1/2$. Il s'ensuit que la valeur du seuil recherchée ϵ^* est comprise dans l'intervalle $[0, 1/2]$. En utilisant l'équation (5.1) nous pouvons évaluer la probabilité d'effacement théorique

associée à chacune des itérations effectuées par le décodeur itératif.

Supposons *a priori* que la probabilité d'effacement du canal ϵ est égale à 0.46. Par conséquent, la probabilité d'effacement théorique à la première itération p_1 est égale à $\epsilon\lambda(1 - \rho(1 - p_0))$, où p_0 prend la valeur ϵ . Dès lors, la valeur p_1 est donnée par l'équation $0.46(1 - (1 - 0.46)^5)^2 = 0.4187$. À la suite de la seconde itération, la probabilité d'effacement p_2 devient égale à $0.46(1 - (1 - 0.4187)^5)^2 = 0.401$. En continuant les calculs, nous pouvons constater qu'à partir de la seizième itération la probabilité d'effacement devient égale à 0.3789 et demeure constante pour toutes les itérations subséquentes effectuées par le décodeur itératif. Nous remarquons donc avec cet exemple, que l'ensemble des codes convolutionnels récurrents définis par la paire de distributions (x^2, x^5) ne permet pas de faire tendre la probabilité d'effacement vers zéro lorsque la probabilité d'effacement associée au canal est égale à 0.46.

Cependant, si nous considérons une probabilité d'effacement initiale égale, par exemple à $\epsilon = 0.4294$, nous pouvons calculer la probabilité d'effacement théorique comme nous venons de le faire précédemment, c'est-à-dire $p_0 = 0.4294, p_1 = 0.3790, \dots, p_{50} = 0.2703, \dots, p_{483} = 0.0017, p_{484} = 0.000029567$. Comme nous pouvons le constater lorsque $\epsilon = 0.4294$, la probabilité d'effacement tend vers zéro au fur et à mesure que le nombre d'itérations augmente. La Figure 5.2 montre l'évolution de la probabilité d'effacement p_ℓ en fonction du nombre d'itérations effectuées par le décodeur itératif pour différentes valeurs de ϵ lorsque le graphe biparti de Tanner est décrit par la paire de distributions (x^2, x^5) . Les résultats indiqués par cette figure nous montrent que la probabilité d'effacement tend vers une constante pour l'ensemble des valeurs ϵ supérieures à 0.4294. Cependant, lorsque la valeur initiale ϵ est inférieure ou égale à 0.4294 nous constatons que p_ℓ tend vers zéro. La valeur du seuil ϵ^* est donc définie comme étant la plus grande valeur numérique ϵ qui permet de faire tendre la probabilité d'effacement vers zéro. Dans cet exemple, cette valeur est égale à 0.4294. On remarque alors que l'ensemble des codes convolutionnels récurrents défini par la paire de distributions (x^2, x^5) peut au mieux

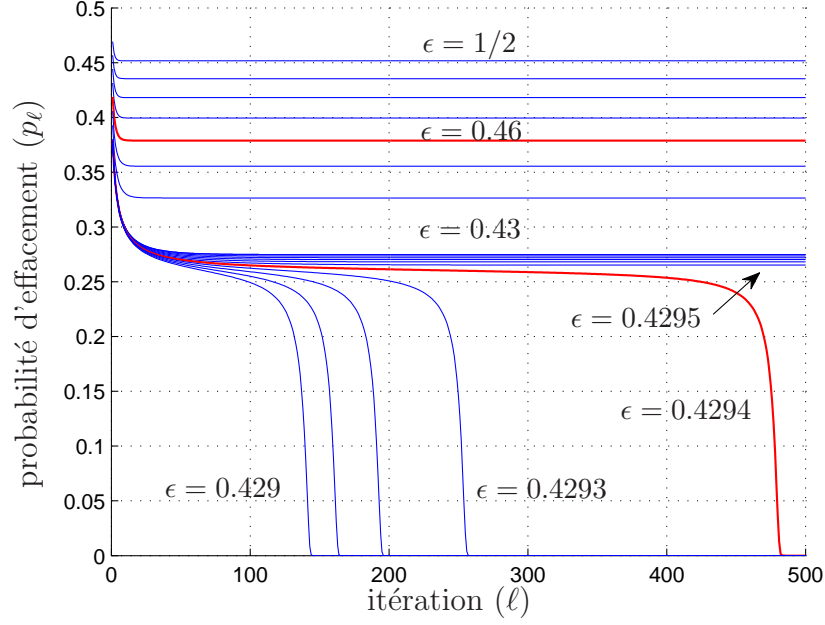


FIGURE 5.2 Évolution de la probabilité d'effacement p_ℓ en fonction du nombre d'itérations ℓ effectuées par le décodeur itératif.

s'approcher à 14.12 % de la capacité de Shannon.

5.1.1 Paires de distributions adaptées aux codes RCDO pour le canal binaire à effacement

La question de recherche à laquelle nous tentons de répondre dans cette section est la suivante. Existe-t-il des paires de distributions adaptées aux codes convolutionnels récurrents doublement orthogonaux qui permettent de s'approcher de la limite de Shannon pour le canal BEC ? L'objectif que nous nous sommes fixés est donc de trouver des paires de distributions (λ, ρ) adaptées aux matrices de contrôle $\mathbf{H}^T(D)$ des codes convolutionnels récurrents de taux de codage $r = b/c$, pour lesquelles la valeur du seuil de convergence de l'algorithme itératif s'approche de la valeur de la capacité de Shannon. Dans cette thèse, nous avons effectué une recherche exhaustive des cas possibles pour des graphes bipartis correspondant aux codes en question et dont le

degré des noeuds variables est inférieur à seize et pour lesquels le taux de codage est égal à $1/2$. C'est-à-dire que nous avons recherché des paires de distributions (λ, ρ) pour des matrices de contrôle $\mathbf{H}^T(D)$ de dimensions $c \times (c-b)$, $4 \leq b \leq 15$ et $c = 2b$. Nous rappelons à la suite du chapitre 2, que les matrices de contrôle qui définissent les codes convolutionnels récurrents doublement orthogonaux sont celles représentées par l'équation (5.3).

$$\mathbf{H}^T(D) = \begin{pmatrix} h_{1,1}D^{\alpha_{1,1}} & \dots & h_{1,(c-b)}D^{\alpha_{1,(c-b)}} \\ h_{2,1}D^{\alpha_{2,1}} & \dots & h_{2,(c-b)}D^{\alpha_{2,(c-b)}} \\ \vdots & & \vdots \\ h_{b,1}D^{\alpha_{b,1}} & \dots & h_{b,(c-b)}D^{\alpha_{b,(c-b)}} \\ h_{b+1,1}D^{\alpha_{b+1,1}} & \dots & h_{b+1,(c-b)}D^{\alpha_{b+1,(c-b)}} \\ \vdots & & \vdots \\ h_{c,1}D^{\alpha_{c,1}} & \dots & h_{c,(c-b)}D^{\alpha_{c,(c-b)}} \end{pmatrix} \quad (5.3)$$

Comme il a été démontré dans (Richardson et Urbanke, 2001a) pour s'approcher de la capacité du canal, les graphes bipartis de Tanner définissant les codes convolutionnels récurrents doivent contenir des noeuds variables de degré au moins égal à deux. Ceci se traduit obligatoirement dans notre recherche des paires de distributions pour lesquelles correspondent une matrice de contrôle $\mathbf{H}^T(D)$ qui possède un certain nombre de lignes ayant au moins deux éléments non-nul. De plus, en considérant la conjecture présentée dans (Luby *et al.*, 2001) nous n'avons limité, le développement du polynôme $\rho(x)$ qu'à deux termes, soit $\rho(x) = \rho_i x^{i-1} + \rho_{i+1} x^i$, où $2 \leq i \leq c$. Les valeurs choisies représentent des valeurs pratiques qui nous ont permis d'effectuer la recherche des graphes bipartis RCDO offrant le meilleur seuil possible dans un temps de recherche raisonnable.

Les Tableaux 5.1 et 5.2 présentent les meilleures paires de distributions que nous avons obtenues suite à notre recherche. Chaque colonne des tableaux indique la meilleure paire de distributions (λ, ρ) que nous avons trouvée pour la taille de la matrice de contrôle indiquée.

TABLEAU 5.1 Paire de distributions adaptée aux codes RCDO de taux de codage $r = b/c$ égal à $1/2$ pour le canal BEC, $4 \leq b \leq 10$. On indique aussi pour chaque paire de distributions, la valeur du seuil ϵ^* , ainsi que la différence entre la capacité de Shannon et la valeur du seuil $\Delta\epsilon$.

$\lambda_i \mid \rho_i$	Dimensions de $\mathbf{H}^T(D) : c \times (c - b)$						
	8×4	10×5	12×6	14×7	16×8	18×9	20×10
λ_2	0.4545	0.4	0.4	0.3810	0.3673	0.3509	0.3385
λ_3		0.1714	0.2371	0.2857	0.3061	0.2632	0.2769
λ_4	0.5455					0.0526	
λ_5		0.4286					
λ_6			0.3429				0.0923
λ_7				0.3333			
λ_8					0.3265		
λ_9						0.3152	0.1385
λ_{10}							0.1538
ρ_5	0.4545	0.1429	0.1429				
ρ_6	0.5455	0.8571	0.8571	1	0.8571	0.6316	0.4615
ρ_7					0.1429	0.3684	0.5385
$ E_{\mathcal{P}} $	22	35	35	42	49	57	65
ϵ^*	0.4632	0.4675	0.4736	0.4785	0.4806	0.4831	0.4849
$\Delta\epsilon$	0.0368	0.0325	0.0264	0.0215	0.0194	0.0169	0.0151

La Figure 5.3 synthétise les résultats importants que nous pouvons dégager des résultats obtenus. Cette figure nous indique les valeurs des seuils obtenues en fonction du degré maximum des noeuds variables dans le protographe des codes convolutionnels récurrents RCDO de taux de codage $1/2$. Le premier constat que nous pouvons observer est qu'une augmentation du degré des noeuds variables permet d'approcher la valeur du seuil de convergence théorique ϵ^* vers la limite de Shannon qui est de $1/2$. Nous pouvons aussi noter que la paire de distributions trouvée, qui offre le meilleur seuil à ce jour, n'est seulement qu'à environ 2% de cette limite. Il est donc naturel de se poser la question suivante. Existe-t-il des paires de distributions adaptées aux codes RCDO qui offrent de meilleurs seuils que ceux trouvés ? La réponse à cette question demeure un problème ouvert. Mais la réponse est probablement positive, car nous n'avons parcouru qu'une petite portion du spectre des solutions possibles étant donné que nous avons limité le nombre de termes qui décrit le polynôme $\rho(x)$.

TABLEAU 5.2 Paire de distributions adaptée aux codes RCDO de taux de codage $r = b/c$ égal à $1/2$ pour le canal BEC, $11 \leq b \leq 15$. On indique aussi pour chaque paire de distributions, la valeur du seuil ϵ^* , ainsi que la différence entre la capacité de Shannon et la valeur du seuil $\Delta\epsilon$.

$\lambda_i \mid \rho_i$	Dimensions de $\mathbf{H}^T(D) : c \times (c - b)$				
	22×11	24×12	26×13	28×14	30×15
λ_2	0.3443	0.3171	0.3076	0.3030	0.2963
λ_3	0.2432	0.2560	0.2637	0.2424	0.25
λ_4	0.0541				
λ_5		0.0610		0.1010	
λ_6	0.0811	0.0732	0.0659		0.1111
λ_7			0.0769	0.0707	0.0648
λ_8					
λ_9					
λ_{10}					
λ_{11}	0.2973				
λ_{12}		0.2927			
λ_{13}			0.2857		
λ_{14}				0.2828	
λ_{15}					0.2778
ρ_6	0.2432	0.1463			
ρ_7	0.7468	0.8537	1	0.9192	0.7778
ρ_8				0.0808	0.2222
$ E^P $	74	82	91	99	108
ϵ^*	0.4868	0.4883	0.4890	0.4897	0.4903
$\Delta\epsilon$	0.0132	0.0117	0.0110	0.0103	0.0097

Nous pouvons aussi ajouter, qu'il est préférable de considérer l'ensemble des codes convolutionnels RCDO dont le graphe biparti est irrégulier. À titre comparatif, nous avons évalué les seuils correspondant aux codes rékursifs dont le graphe biparti est régulier. Les résultats obtenus sont rapportés au Tableau 5.3. Comme nous pouvons le constater l'ensemble des codes RCDO réguliers qui s'approchent le plus près de la limite de Shannon est celui cité à l'exemple 5.1, où $\lambda(x) = x^2$ et $\rho(x) = x^5$. De plus, nous pouvons remarquer que tous les seuils associés aux codes RCDO irréguliers des Tableaux 5.1 et 5.2 offrent un gain de codage substantiel en comparaison avec la paire de distributions régulière étudiée à l'exemple 5.1. D'où l'importance du choix de la

paire de distributions (λ, ρ) des arêtes dans le protographe d'un code.

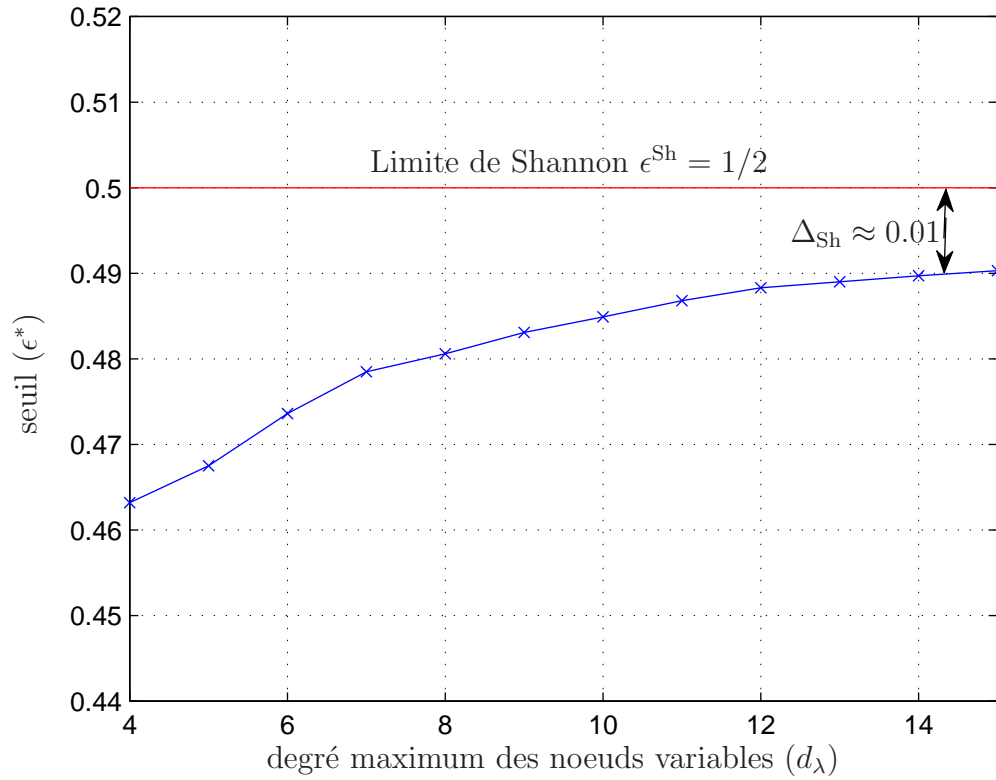


FIGURE 5.3 Meilleurs seuils trouvés ϵ^* en fonction du degré maximum alloué aux noeuds variables pour les codes RCDO de taux de codage $r = 1/2$ pour le cas du canal binaire à effacement.

TABLEAU 5.3 Paire de distributions adaptée aux codes RCDO réguliers de taux de codage $r = b/c$ égal à $1/2$ pour le canal BEC. Aussi indiqué pour chaque paire de distributions, la valeur du seuil ϵ^* , le nombre d'arêtes composant le protographe $|E^P|$ ainsi que la différence entre la capacité de Shannon et la valeur du seuil $\Delta\epsilon$.

$(x^{d_\lambda-1}, x^{d_\rho-1})$	seuil (ϵ^*)	$\Delta\epsilon$ ($\epsilon^{\text{Sh}} - \epsilon^*$)
(x^2, x^5)	0.4294	0.0706
(x^3, x^7)	0.3834	0.1166
(x^4, x^9)	0.3416	0.1584

5.1.2 Résultats de simulations des codes RCDO pour le canal binaire à effacement

Dans cette section, nous présentons les résultats de simulations par ordinateur de la performance d'erreur des codes convolutionnels récurrents doublement orthogonaux de taux de codage $1/2$, lorsqu'un canal binaire à effacement est considéré pour modéliser la transmission de l'information. Les codes convolutionnels doublement orthogonaux générés vérifient tous une des paires de distributions obtenues au Tableau 5.1.

Notons que les quatre codes dont nous présentons les résultats sont tous de taux de codage $r = b/c$ où $c = 2b$ et $b \in \{4, 6, 8, 10\}$. De plus, pour alléger le texte, toutes les matrices de contrôle que nous avons générées pour ces quatre codes convolutionnels récurrents sont présentées à l'AnnexeC.

Exemple 5.2- À titre d'illustration, l'équation (5.4) présente la matrice de contrôle du code RCDO de taux de codage $r = 4/8$, qui possède huit lignes et quatre colonnes. La paire de distributions qui correspond à cette matrice est celle définie par les polynômes $\lambda(x) = 0.4545x + 0.5454x^3$ et $\rho(x) = 0.4545x^4 + 0.5455x^5$.

$$\mathbf{H}^T(D) = \begin{pmatrix} D^{100} & D^{86} & D^{55} & D^{83} \\ D^{51} & D^{11} & 0 & 0 \\ D^{74} & 0 & 0 & D^{15} \\ D^{37} & D^{78} & D^{82} & D^{19} \\ 1 & 0 & D^{41} & 0 \\ D^{34} & 1 & D^{17} & D^{54} \\ 0 & D^{23} & 1 & 0 \\ 0 & 0 & D^{94} & 1 \end{pmatrix} \quad (5.4)$$

En fait, les coefficients de ces polynômes correspondent aux valeurs indiquées dans la première colonne du Tableau 5.1. Comme nous pouvons le constater pour cette paire de distributions, le protographe associé au code possède vingt-deux arêtes ce qui correspond au nombre d'éléments non nuls de la matrice $\mathbf{H}^T(D)$ du code RCDO. Pour générer une matrice de contrôle qui répond à cette paire de distributions, il suffit de générer une matrice de contrôle qui possède n_j^v lignes possédant j éléments non nuls et n_k^c colonnes avec k éléments non-nul, $2 \leq j \leq c$ et $1 \leq k \leq (c-b)$. Les valeurs n_j^v et n_k^c sont définies par :

$$n_j^v = \frac{|E_{\mathcal{P}}|\lambda_j}{j} \quad \text{et} \quad n_k^c = \frac{|E_{\mathcal{P}}|\rho_k}{k} \quad (5.5)$$

Une matrice de contrôle qui correspond à cette paire de distributions est celle de l'équation (5.6). Pour cette matrice, il y a : $n_2^v = 5$ lignes possédant deux éléments non nuls, et $n_4^v = 3$ lignes possédant quatre éléments non nuls. De même, il y a $n_5^c = 2$ colonnes qui possèdent cinq éléments non nuls, et $n_6^c = 2$ colonnes avec six éléments non nuls. Cette matrice de contrôle, sous cette forme ne répond pas à la définition de la double orthogonalité des codes convolutionnels récursifs présentée au chapitre précédent. Pour obtenir un code RCDO, il suffit de s'assurer que les valeurs entières $\alpha_{m,n}$ de la matrice de l'équation (5.6) répondent à la définition des codes RCDO.

$$\mathbf{H}^T(D) = \begin{pmatrix} D^{\alpha_{1,1}} & D^{\alpha_{1,2}} & D^{\alpha_{1,3}} & D^{\alpha_{1,4}} \\ D^{\alpha_{2,1}} & D^{\alpha_{2,2}} & 0 & 0 \\ D^{\alpha_{3,1}} & 0 & 0 & D^{\alpha_{3,4}} \\ D^{\alpha_{4,1}} & D^{\alpha_{4,2}} & D^{\alpha_{4,3}} & D^{\alpha_{4,4}} \\ 1 & 0 & D^{\alpha_{5,3}} & 0 \\ D^{\alpha_{6,1}} & 1 & D^{\alpha_{6,3}} & D^{\alpha_{6,4}} \\ 0 & D^{\alpha_{7,2}} & 1 & 0 \\ 0 & 0 & D^{\alpha_{8,3}} & 1 \end{pmatrix} \quad (5.6)$$

Une solution possible est la matrice de contrôle présentée à l'équation (5.4). En se référant au tableau 5.1, pour ce code RCDO le seuil théorique de convergence de l'algorithme de décodage est donc égal à 0.4632.

À la Figure 5.4, nous présentons les performances de quatre codes RCDO de taux de codage 4/8, 6/12, 8/16 et 10/20 auxquels correspondent les seuils indiqués au tableau 5.4.

TABLEAU 5.4 Valeurs des seuils correspondant aux différents codes de la Figure 5.4.

r	mémoire (m_s)	seuil (ϵ^*)	Δ_{Sh}
4/8	100	0.4632	0.0368
6/12	149	0.4736	0.0264
8/16	199	0.4806	0.0194
10/20	300	0.4849	0.0151

À titre de comparaison, nous avons aussi simulé les performances d'effacement d'un code convolutionnel récursif systématique de taux de codage 4/8 pour lequel les positions des connexions du codeur convolutionnel ont été générées de façon aléatoire. Pour assurer une comparaison équitable avec le code RCDO de taux de codage 4/8, nous avons utilisé la matrice de contrôle $\mathbf{H}^T(D)$ présentée à l'équation (5.6), mais sans exiger que les valeurs $\alpha_{m,n}$ rencontrent la définition de la double orthogonalité. De plus, nous nous sommes assurés que la valeur de la mémoire du plus grand registre à décalage m_s soit identique et égale à 100 pour les deux codeurs convolutionnels récursifs. Comme nous l'avons présenté au chapitre précédent, cette précaution fait en sorte que les deux décodeurs itératifs introduisent la même latence de décodage.

Les résultats de simulations présentés à la Figure 5.4, à la quarantième itération, nous montrent clairement l'avantage d'imposer la double orthogonalité. Le code RCDO de taux de codage $r = 4/8$ offre de meilleures performances que celles du code pour lequel la position des connexions fut générée de façon aléatoire. Ceci peut s'expliquer par l'introduction de plus petits cycles dans la structure du graphe biparti associée au code convolutionnel récursif aléatoire.

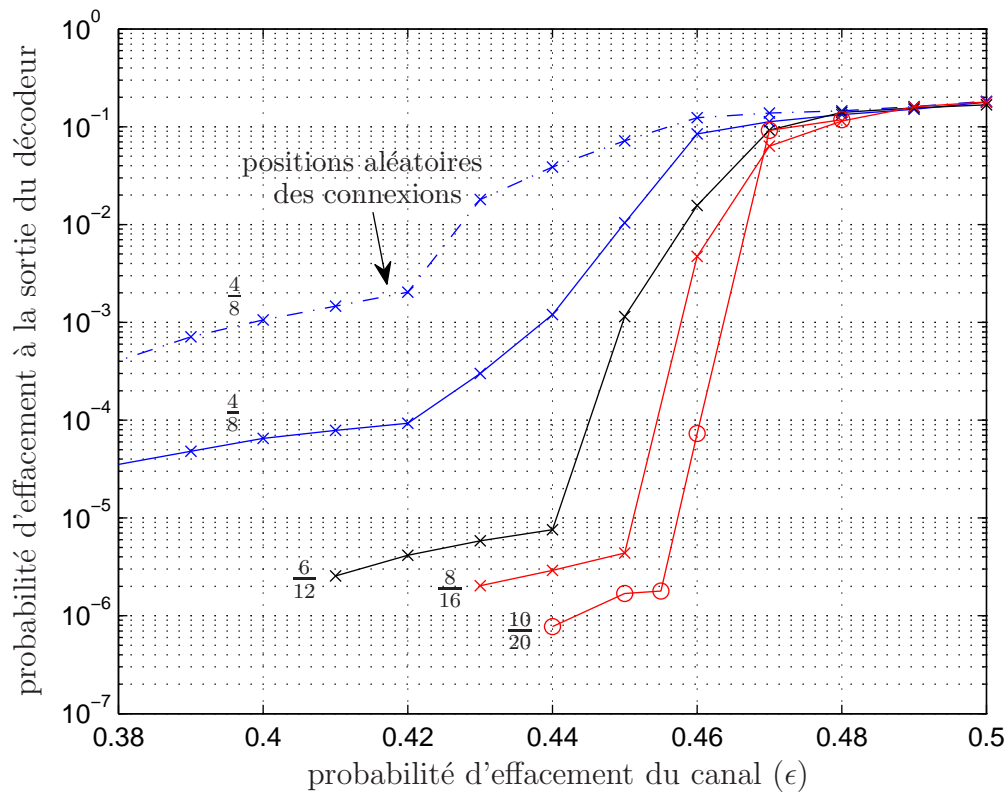


FIGURE 5.4 Résultats de simulations pour quatre codes RCDO de taux de codage 1/2 pour le canal binaire à effacement. Le nombre d'itérations effectués par le décodeur itératif égal à quarante.

De plus, nous pouvons aussi constater qu'il existe un écart entre les différents seuils obtenus et les résultats simulés. Pour tous les codes présentés, l'écart le plus grand qui sépare la valeur du seuil théorique à la valeur du plancher d'effacement est d'environ 10%. Cet écart peut s'expliquer par le fait que les codes RCDO contiennent des cycles de longueur 10, mais que la valeur du seuil calculée théoriquement fait abstraction de ces cycles dans le graphe biparti des codes. Par exemple, le code RCDO

de taux de codage $r = 10/20$ offre des performances qui se situent à environ 9% de la limite de Shannon tandis que le résultat théorique qui est associé à cette paire de distribution est à environ 3% de cette limite. Néanmoins, même s'il persiste un écart entre les résultats théoriques et les expériences numériques, il demeure que les seuils trouvés restent un bon indicateur de la convergence des codes RCDO. C'est-à-dire, qu'il devient maintenant possible de classer les performances du décodeur itératif des codes RCDO en fonction du seuil trouvé théoriquement. En effet, on constate que les résultats obtenus par simulations suivent la même tangente que les résultats théoriques, c'est-à-dire que les codes RCDO irréguliers qui possèdent un graphe biparti dont le degré des noeuds variables est élevé sont plus performants que ceux qui possèdent un graphe biparti dont le degré des noeuds variables est faible.

Les résultats obtenus nous permettent aussi de remarquer un autre phénomène qui est celui du plancher d'effacement. Pour tous les codes RCDO présentés, un plancher d'effacement apparaît. La cause de ce plancher provient de certains sous-graphes induits qui composent le graphe biparti des codes. Ces sous-graphes induits sont appelés ensembles d'arrêt, *stopping sets*. En fait, ces ensembles d'arrêt empêchent l'algorithme itératif de converger vers une probabilité d'effacement qui tend vers zéro, laissant un certain nombre de symboles effacés. Nous concluons la discussion sur ce sujet, en mentionnant que nos résultats laissent croire qu'augmenter la valeur m_s ainsi que les dimensions de la matrice de contrôle $\mathbf{H}^T(D)$ peuvent permettre la diminution de ce plancher d'effacement. Comme l'étude des ensembles d'arrêt ne fait pas partie de cette thèse, nous référons donc le lecteur intéressé par ce sujet aux articles suivants (Orlitsky *et al.*, 2005) et (Richter et Hof, 2006).

5.2 Seuils de convergence des codes RCDO pour le canal binaire symétrique à bruit additif blanc et gaussien

Dans cette section, nous considérons l'utilisation de codes RCDO, mais cette fois-ci lorsque la transmission de l'information s'effectue dans un canal binaire symétrique à bruit additif blanc et gaussien (AWGN), où la sortie du canal est non quantifiée. Ce canal de communication représente bien un canal de communication satellitaire, par exemple. Tout comme pour le canal BEC, nous cherchons à savoir s'il existe des paires de distributions (λ, ρ) adaptées aux protographes des codes RCDO qui permettent de s'approcher de la limite de Shannon. Comme nous l'avons présenté au Tableau 2.1 du chapitre 2, la capacité de Shannon pour le canal AWGN est une valeur qui varie en fonction du taux de codage des codes utilisés. Notre recherche de paires de distributions pour ce canal se limitera plus particulièrement aux codes RCDO de taux de codage $1/3$, $1/2$ et $2/3$.

Contrairement au canal binaire à effacement, l'analyse de l'évolution de la probabilité d'erreur pour le canal AWGN non quantifié est beaucoup plus difficile. L'algorithme utilisé pour évaluer la valeur du seuil de convergence de l'algorithme itératif de décodage se retrouve dans (Richardson et Urbanke, 2001a) et (Richardson *et al.*, 2001). Cet algorithme permet d'évaluer la distribution de probabilité associée aux différents messages échangés dans le graphe biparti des codes lors du décodage. Sans décrire les menus détails de l'algorithme en question, nous énonçons les différentes hypothèses ainsi que les équations qui permettent d'évaluer la valeur du seuil qui correspond à l'ensemble des graphes bipartis *sans cycle* définis par une paire de distributions (λ, ρ) .

5.2.1 Paires de distributions adaptées aux codes RCDO pour le canal additif à bruit blanc et gaussien

Comme nous l'avons énoncé au chapitre 2, l'algorithme utilisé pour décoder les symboles d'information utilise comme mesure le logarithme du rapport de vraisemblance (LRV) pour propager sur les arêtes du graphe biparti des messages qui servent, d'une itération à l'autre, à estimer les différents symboles d'information. En considérant

un récepteur optimal constitué de filtres adaptés aux signaux transmis, et un bruit additif blanc et gaussien, on peut montrer que la valeur à la sortie du filtre adapté $y_i^{(m)}$ correspond à la somme (Lin et Costello, 2004) :

$$y_i^{(m)} = x_i^{(m)} + n_i^{(m)} \quad (5.7)$$

où $x_i^{(m)}$ dépend de la modulation antipodale choisie, $x_i^{(m)} \in \{-1, +1\}$, et où $n_i^{(m)}$ représente une variable aléatoire qui possède une densité de probabilité gaussienne de moyenne nulle et de variance σ^2 . Par conséquent, la valeur $y_i^{(m)}$ représente une variable aléatoire qui possède une densité de probabilité f_Y gaussienne de moyenne $x_i^{(m)}$ et de variance σ^2 . Si nous considérons une transmission équiprobable des signaux $x_i^{(m)}$, alors les valeurs LRV utilisées par le décodeur itératif sont représentées par l'équation suivante (Berrou *et al.*, 1993) :

$$\begin{aligned} L(y_i^{(m)}) &= \ln \left(\frac{f_Y(y_i^{(m)} | x_i^{(m)} = +1)}{f_Y(y_i^{(m)} | x_i^{(m)} = -1)} \right) \\ &= \ln \left(\frac{\exp(-\frac{1}{2} \frac{(y_i^{(m)} - 1)^2}{\sigma^2})}{\exp(-\frac{1}{2} \frac{(y_i^{(m)} + 1)^2}{\sigma^2})} \right) \\ &= \frac{2}{\sigma^2} y_i^{(m)} \end{aligned} \quad (5.8)$$

Ces LRV sont donc des variables aléatoires gaussiennes, car $y_i^{(m)} \sim \mathcal{N}(\pm 1, \sigma^2)$. Pour analyser l'évolution de la probabilité d'erreur durant le processus de décodage, nous considérons la transmission du mot de code nul, $\mathbf{v} = \mathbf{0}$. C'est-à-dire, qu'en considérant la règle de correspondance :

$$x_i^{(m)} = 1 - 2v_i^{(m)} \quad (5.9)$$

les signaux à la sortie du modulateur $x_i^{(m)}$ prennent toujours la valeur positive +1. Sous-cette hypothèse, l'espérance mathématique de la variable aléatoire $L(y_i^{(m)})$ est donc :

$$\begin{aligned} \mathbf{E}\{L(y_i^{(m)})\} &= \mathbf{E}\{2 \frac{y_i^{(m)}}{\sigma^2}\} \\ &= \frac{2}{\sigma^2} \mathbf{E}\{y_i^{(m)}\} \\ &= \frac{2}{\sigma^2} \end{aligned} \quad (5.10)$$

et la variance de la variable aléatoire $L(y_i^{(m)})$ est :

$$\begin{aligned}
 \text{var} \left(L(y_i^{(m)}) \right) &= \text{var} \left(\frac{2}{\sigma^2} y_i^{(m)} \right) \\
 &= \frac{4}{\sigma^4} \text{var} \left(x_i^{(m)} + n_i^{(m)} \right) \\
 &= \frac{4}{\sigma^4} \text{var} \left(n_i^{(m)} \right) \\
 &= \frac{4}{\sigma^2}
 \end{aligned} \tag{5.11}$$

Les LRV $L(y_i^{(m)})$ peuvent donc être représentés par des variables aléatoires possédant une densité de probabilité gaussienne de moyenne $2/\sigma^2$ et de variance $4/\sigma^2$ décrit par l'équation :

$$f_L(x) = \frac{\sigma}{2\sqrt{2\pi}} \exp \left(-\frac{\sigma^2(x - \frac{2}{\sigma^2})^2}{8} \right) \tag{5.12}$$

c'est-à-dire, $L(y_i^{(m)}) \sim \mathcal{N}(2/\sigma^2, 4/\sigma^2)$. En notant l'énergie d'un bit par E_b et en considérant un code correcteur d'erreur de taux de codage égal à r ainsi qu'un bruit blanc dont la densité spectrale bilatérale est égal à $\frac{N_0}{2}$ il s'ensuit que $\sigma^2 = \frac{1}{2r} \frac{E_b}{N_0}$ (Lin et Costello, 2004).

L'analyse de l'algorithme présenté dans (Richardson *et al.*, 2001), nous permet d'évaluer numériquement la densité de probabilité des messages des noeuds variables à l'itération ℓ pour un code qui possède un graphe biparti défini par une paire de distributions (λ, ρ) . La densité de probabilité correspondant aux messages des noeuds variables peut être obtenue selon la relation itérative suivante (Richardson *et al.*, 2001) :

$$f_L^{(\ell)}(x) = \lambda \left(\Gamma^{-1} \left(\rho(\Gamma(f_L^{(\ell-1)}(x))) \right) \right) \otimes f_L^{(0)}(x) \tag{5.13}$$

où,

$$\Gamma^{-1} \left(\rho(\Gamma(f_L^{(\ell-1)}(x))) \right) = \Gamma^{-1} \left(\sum_i \rho_i \left(\Gamma(f_L(x)^{(\ell-1)}) \right)^{\otimes i-1} \right) \tag{5.14}$$

et

$$\lambda \left(\Gamma^{-1} \left(\rho(\Gamma(f_L^{(\ell-1)}(x))) \right) \right) = \sum_i \lambda_i \left(\Gamma^{-1} \left(\rho(\Gamma(f_L^{(\ell-1)}(x))) \right) \right)^{\otimes i-1} \tag{5.15}$$

où \otimes dénote l'opération de convolution. De plus, $f_L^{(0)}(x)$ représente la densité de probabilité initiale associée aux messages $m_{v_i^{(m)}}^{(0)}$ utilisés à l'entrée du décodeur itératif.

Cette densité de probabilité correspond à la densité de probabilité de la variable aléatoire $L(y_i^{(m)})$ donnée par l'équation (5.12). Les fonctions Γ et Γ^{-1} dépendent de l'opérateur utilisé pour effectuer la mise à jour des messages au niveau des noeuds de contraintes et sont définies dans (Richardson et Urbanke, 2008). En émettant l'hypothèse, que le mot de code nul, $\mathbf{v} = \mathbf{0}$, est celui généré par le code, il s'en suit que la probabilité d'erreur moyenne par bit peut être exprimée par (Richardson *et al.*, 2001) :

$$\bar{p}_b^{(\ell)} = \int_{-\infty}^0 f_L^{(\ell)}(x) dx \quad (5.16)$$

Exemple 5.3- Cet exemple sert à illustrer schématiquement le principe associé aux équations (5.13) à (5.16). La Figure 5.5 correspond à l'arbre de calculs utilisé pour effectuer l'estimation d'un des noeuds variables qui compose un graphe biparti sans cycle défini par la paire de distribution (λ, ρ) . Si nous considérons la transmission du mot de code nul, i.e. $\mathbf{v} = \mathbf{0}$, alors la densité de probabilité $f_L^{(0)}$ correspondant aux noeuds variables à l'entrée du décodeur est représentée par une distribution gaussienne de moyenne $2/\sigma^2$ et de variance $4/\sigma^2$. Cette densité de probabilité, qui possède une moyenne positive, est associée aux feuilles de l'arbre. Ces valeurs sont donc transmises vers un noeud de contrainte qui transmettra vers un noeud variable un message dont la densité de probabilité est donnée par l'équation (5.14). Les variables qui reçoivent les messages provenant des noeuds de contraintes possèdent une densité de probabilité donnés par l'équation (5.13). À la fin des \mathcal{L} itérations, la variable estimée est donc représentée par un message ayant une densité de probabilité égale à $f_L^{(\mathcal{L})}$.

Comme nous supposons la transmission du mot de code nul, une erreur de décodage est effectuée si la variable décodée après \mathcal{L} itérations prend la valeur binaire 1. La probabilité associée à cet événement est donc décrite par l'équation (5.16). Cette équation représente en fait l'aire sous la courbe qui se trouve à gauche de l'axe y de la densité de probabilité du message qui correspond à la variable estimée, soit la racine de l'arbre.

Étant donné que la densité de probabilité associée aux messages initiaux $f_L^{(0)}$ dépend uniquement de la variable σ^2 , nous nous intéressons donc à

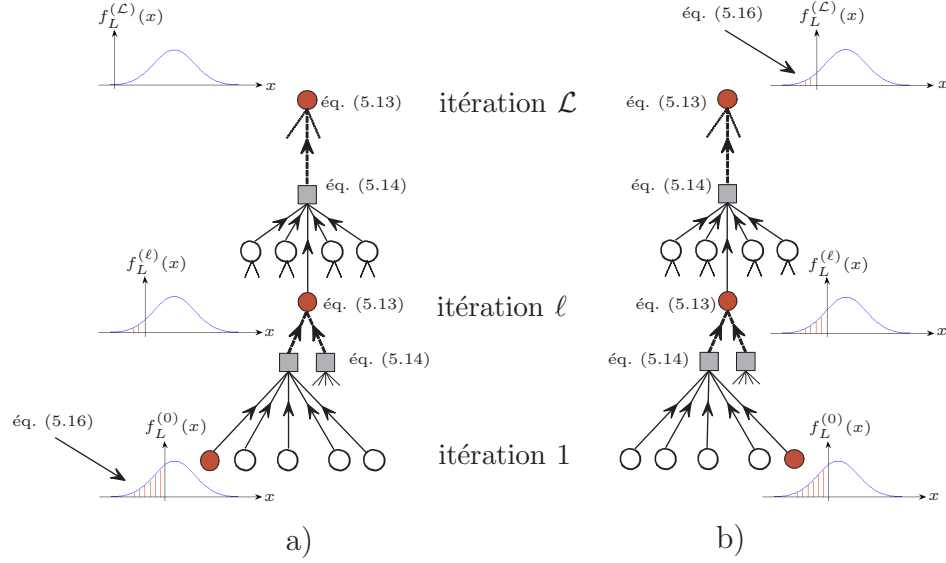


FIGURE 5.5 Exemple schématique expliquant le fonctionnement de l'algorithme de l'évolution de la densité de la probabilité pour un graphe ayant une paire de distribution (λ, ρ) . a) Cas pour lequel la probabilité d'erreur tend vers zéro après avoir effectué L itérations de décodage. b) Cas pour lequel la probabilité d'erreur tend vers une constante après avoir effectué L itérations de décodage.

la valeur σ^2 minimum qui permet de faire tendre vers zéro la probabilité d'erreur calculée à l'équation (5.16). La valeur σ^2 correspond à $\frac{1}{2r} \frac{E_b}{N_0}$ et par conséquent, nous nous intéressons plus particulièrement à l'ensemble des paires de distributions (λ, ρ) pour lesquelles la valeur de $\frac{E_b}{N_0}$ tend vers la limite de Shannon lorsque l'équation (5.16) tend vers 0. La plus petite valeur de $\frac{E_b}{N_0}$ qui permet la convergence de l'algorithme itératif est donc le seuil de convergence théorique $(\frac{E_b}{N_0})^*$. Cette valeur possède comme borne inférieure, la limite de Shannon.

Les seuils correspondant aux ensembles de codes RCDO *réguliers* de taux de codage $r = 1/2$, dont les graphes bipartis de Tanner possèdent tous des noeuds variables et de contraintes de degrés constants égaux à d_λ et d_ρ , sont présentés au tableau 5.5. Ce tableau montre qu'il n'y a aucune paire de distributions régulières

$(x^{d_\lambda-1}, x^{d_\rho-1})$ qui permet la convergence de l'algorithme itératif à passage de messages à moins de 0.92 dB de la limite de Shannon. Les seuils obtenus correspondent aussi aux seuils obtenus par les codes LDPC en blocs *réguliers* lorsque ces derniers possèdent des graphes bipartis de Tanner ayant comme paires de distributions $(x^{d_\lambda-1}, x^{d_\rho-1})$.

À la Figure 2.7 de la section 2.3 du chapitre 2, nous avons présenté les performances d'erreur des deux codes RCDO présentés dans (Cardinal, 2001). Avec la représentation adoptée au chapitre 3, nous pouvons représenter la matrice de contrôle $\mathbf{H}^T(D)$ du code RCDO-A par l'équation (5.17).

$$\mathbf{H}^T(D) = \begin{pmatrix} 1 & D^{299} & 1 & D^{142} & D^{1135} \\ 1 & D^{320} & D^{146} & D^{107} & D^{794} \\ 1 & D^{679} & D^5 & D^{977} & D^{172} \\ D^{87} & D^{333} & 1 & D^{106} & D^{310} \\ 1 & D^{418} & D^{60} & D^{123} & D^{99} \\ 1 & D^{945} & D^{210} & D^{90} & D^{92} \\ D^{1135} & 1 & D^{552} & D^{192} & 1 \\ D^{39} & D^{296} & 1 & D^{1003} & D^{97} \\ D^{387} & D^{198} & D^{193} & 1 & 1 \\ D^{132} & D^{208} & D^{1073} & D^3 & 1 \end{pmatrix} \quad (5.17)$$

Étant donné que cette matrice possède cinq éléments non nuls par ligne et dix éléments non nuls par colonne, il s'ensuit que le protographe associé à ce code est régulier et chaque noeud variable du protographe possède un degré d_λ égal à cinq et chaque noeud de contraintes possède un degré d_ρ égal à dix. Comme nous pouvons le constater à la Figure 2.7 du chapitre 2, le code RCDO-A offre des performances d'erreur qui tendent vers la valeur zéro lorsque le rapport $\frac{E_b}{N_0}$ est supérieur à 2.25 dB. En se référant au Tableau 5.5, on constate que le code RCDO-A n'est qu'à 0.2 dB du seuil de convergence théorique associé aux protographes réguliers ayant la paire de distributions (x^4, x^9) . Avec la représentation adoptée, nous sommes en mesure pour la première fois, de décrire les performances asymptotiques des codes RCDO présentés dans (Cardinal, 2001).

Nous rappelons que le code RCDO-B présenté à la Figure 2.7 représente le code RCDO-A pour lequel l'auteur de (Cardinal, 2001) a éliminé certaines connexions au hasard du codeur RCDO-A. Comme nous l'avons décrit au chapitre 2, le fait d'éliminer

TABLEAU 5.5 Seuils correspondant aux ensembles de codes RCDO réguliers de taux de codage $1/2$ possédant des graphes bipartis décrit par la paire de distribution $(x^{d_\lambda-1}, x^{d_\rho-1})$.

(d_λ, d_ρ)	seuil (dB)	Δ_{Sh} (dB)
(3,6)	1.11	0.92
(4,8)	1.54	1.35
(5,10)	2.01	1.82

des connexions fait en sorte que la matrice de contrôle $\mathbf{H}^T(D)$ de l'équation (5.17) n'est plus constituée que d'éléments non nuls. Ce faisant, le protographe qui représente la matrice de contrôle $\mathbf{H}^T(D)$ du code RCDO-B est un protographe irrégulier. Cette observation va dans le sens des travaux de (Luby *et al.*, 1997) qui montrent que les graphes bipartis des codes LDPC *irréguliers* offrent de meilleurs résultats que ceux obtenus avec les codes LDPC possédant un graphe de Tanner régulier.

Cette observation nous conduit donc à la recherche de paires de distributions (λ, ρ) adaptées aux codes convolutionnels doublement orthogonaux de taux de codage $r = b/c$. Suite aux résultats présentés dans (Richardson et Urbanke, 2001a) et pour limiter notre recherche, nous avons uniquement recherché des paires de distributions pour lesquelles la distribution $\rho(x)$ peut s'écrire sous la forme $\rho(x) = \rho_i x^{i-1} + \rho_{i+1} x^i$ avec $2 \leq i \leq (c-1)$. La Figure 5.6 indique les différentes valeurs de seuils que nous avons trouvées en fonction du degré maximum des noeuds variables alloués dans le protographe des codes RCDO de taux de codage $1/3$, $1/2$ et $2/3$ (Roy *et al.*, 2010). Les trois courbes de la figure représentent les différentes valeurs des seuils présentées aux Tableaux 5.6, 5.7 et 5.8. Nous pouvons remarquer que les meilleurs seuils trouvés, pour les trois taux de codage, sont à moins de 0.5 dB de la limite de Shannon. Pour le cas du taux de codage $1/2$, la paire de distributions irrégulière qui offre le meilleur seuil théorique permet un gain de codage supplémentaire d'environ 0.6 dB par rapport à la valeur du meilleur seuil pour un code RCDO régulier. Nous avons aussi indiqué les différents seuils obtenus dans (Richardson *et al.*, 2001) avec des codes LDPC en blocs de taux de codage $r = 1/2$. Comme nous pouvons le constater, les seuils obtenus pour les codes RCDO *irréguliers* sont supérieurs d'environ 0.1 dB par rapport à ceux des codes LDPC en blocs irréguliers. Néanmoins, même si théoriquement les matrices de contrôle pour les codes RCDO offrent de moins bons résultats que celles

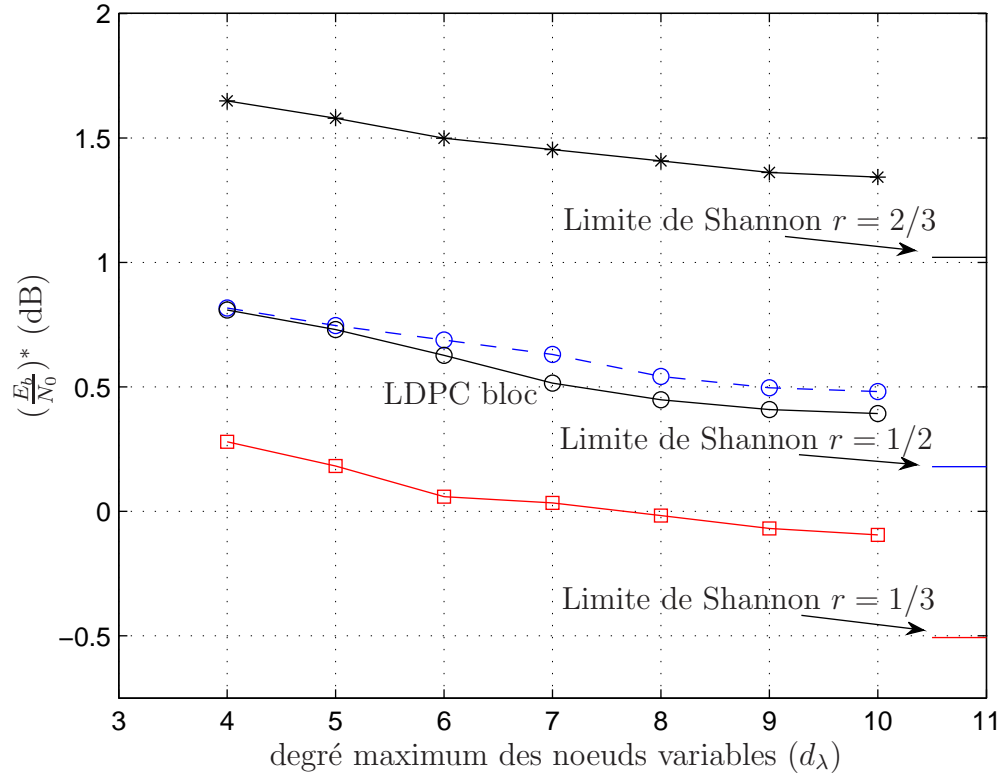


FIGURE 5.6 Meilleurs seuils trouvés en fonction du degré maximum alloué aux noeuds variables pour les codes RCDO de taux de codage $r = 1/3, 1/2$ et $2/3$ pour le cas du canal AWGN.

des codes LDPC irréguliers présentés dans (Richardson *et al.*, 2001), il n'en reste pas moins que la taille des matrices de contrôle des codes RCDO est de beaucoup inférieure à la dimension des matrices de contrôle des codes LDPC en blocs, qui est de 1000000×500000 . La réduction de la taille de la matrice de contrôle se traduit en un avantage majeur lorsqu'il faut éliminer les cycles dans le graphe biparti du code. De plus, les matrices de contrôle des codes RCDO sont structurées ce qui n'est pas nécessairement le cas pour les matrices de contrôle associées aux codes LDPC en blocs. Généralement, l'encodage des codes RCDO est beaucoup plus simple que celui des codes LDPC en blocs non structurés. Cependant, il existe des méthodes de construction des codes LDPC en blocs qui permettent d'incorporer une structure au sein du code. Souvent ces codes sont appelés des codes LDPC *quasi-cyclique* en bloc. Nous pouvons retrouver de tels exemples de constructions dans (Tanner *et al.*, 2004).

À partir de l'analyse sur la complexité effectuée au chapitre 3, nous pouvons analyser la complexité de calculs associée aux meilleures paires de distributions (λ, ρ) des codes RCDO qui se retrouvent aux Tableaux 5.6, 5.7 et 5.8. D'une part, comme nous l'avons mentionné au chapitre 3, la complexité liée aux calculs de l'information extrinsèque dépend du nombre d'arêtes $|E_{\mathcal{P}}|$ composant le protographe du code RCDO ainsi que de la paire de distributions (λ, ρ) correspondant au protographe. Nous avons montré au chapitre précédent, qu'à chaque instant, un processeur doit effectuer un nombre d'additions de nombres réels qui est égal à $|E_{\mathcal{P}}|\lambda'(1)$. Nous avons représenté à la Figure 5.7, le nombre d'additions de nombres réels requis par un processeur en fonction du seuil obtenu pour chacune des paires de distributions indiquées aux Tableaux 5.6, 5.7 et 5.8. Chaque point sur cette figure correspond à une paire de distributions (λ, ρ) qui se retrouve dans l'un de ces tableaux. Nous pouvons remarquer que les codes RCDO ayant des protographes irréguliers possèdent un seuil asymptotique de décodage qui s'approche de la limite de Shannon, mais ce seuil est obtenu au prix d'une croissance substantielle de la complexité de calculs.

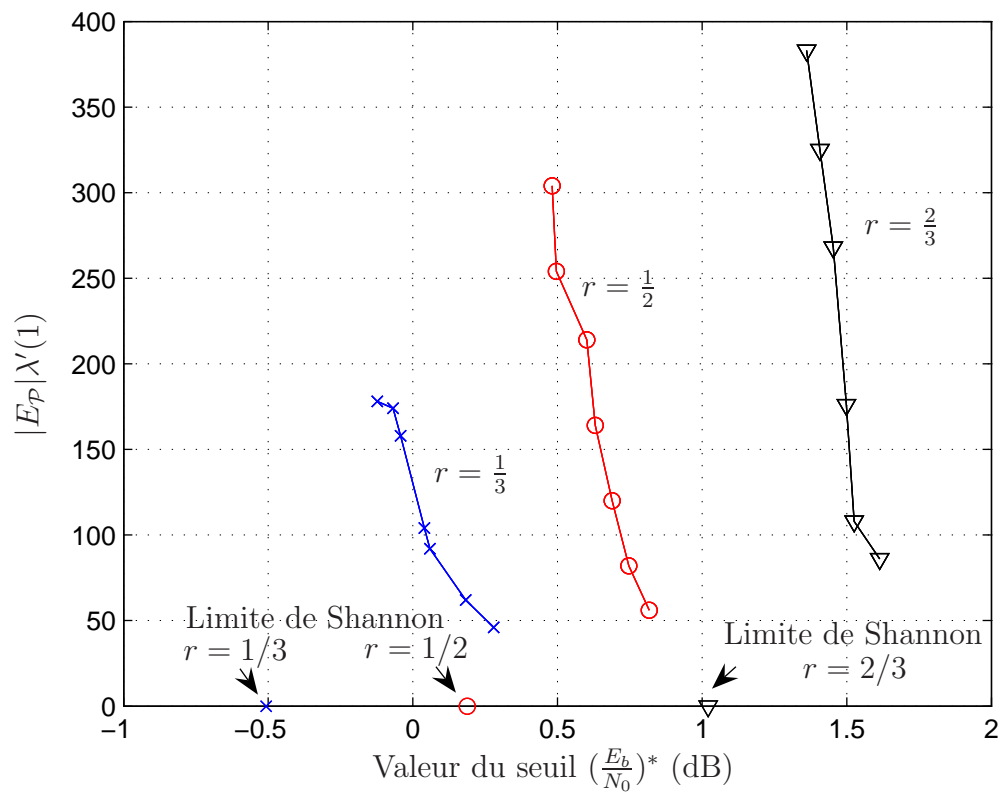


FIGURE 5.7 Nombre d'additions $|E_P|\lambda'(1)$ requises par itération en fonction du seuil associé aux paires de distributions (λ, ρ) des Tableaux 5.6, 5.7 et 5.8.

TABLEAU 5.6 Paires de distributions adaptées aux codes convolutionnels récurrents doublement-orthogonaux de taux de codage $r = b/c$ égal à $1/3$, $b \in \{3, 4, 5\}$, offrant de bons seuils de convergence pour le canal AWGN. Pour chaque paire de distribution, le tableau indique la valeur du seuil $(\frac{E_b}{N_0})^*$ en decibel ainsi que le nombre d'arêtes $|E_{\mathcal{P}}|$ constituant le protographe.

$\lambda_i \mid \rho_i$	Dimensions de $\mathbf{H}^T(D) : c \times (c - b)$						
	9×6	9×6	12×8	12×8	15×10	15×10	15×10
λ_2	0.4167	0.3846	0.4	0.3889	0.3404	0.3333	0.3333
λ_3	0.25	0.2308	0.2571	0.25	0.3191	0.3125	0.3125
λ_4	0.3333						
λ_5		0.3846					
λ_6			0.3429	0.1667			
λ_7				0.1944			0.1458
λ_8					0.3404	0.1667	
λ_9						0.1875	
λ_{10}							0.2083
ρ_4	1	0.6154	0.5714	0.4444	0.2553	0.1667	0.1667
ρ_5		0.3846	0.4286	0.5556	0.7447	0.8333	0.8333
$ E_{\mathcal{P}} $	24	26	35	36	47	48	48
$(\frac{E_b}{N_0})^*$	0.2792	0.1827	0.0590	0.0398	-0.0426	-0.0684	-0.1231

TABLEAU 5.7 Paires de distributions adaptées aux codes convolutionnels récurrents doublement-orthogonaux de taux de codage $r = b/c$ égal à $1/2$, $b \in \{4, 5, 6, 7, 8, 9, 10\}$, offrant de bons seuils de convergence pour le canal AWGN. Pour chaque paire de distribution, le tableau indique la valeur du seuil $(\frac{E_b}{N_0})^*$ en decibel ainsi que le nombre d'arêtes $|E_{\mathcal{P}}|$ constituant le protographe.

$\lambda_i \mid \rho_i$	Dimensions de $\mathbf{H}^T(D) : c \times (c - b)$						
	8×4	10×5	12×6	14×7	16×8	18×9	20×10
λ_2	0.3333	0.3226	0.3077	0.2979	0.2909	0.2903	0.2857
λ_3		0.1935	0.2308	0.2553	0.2727	0.2903	0.2571
λ_4	0.6667						0.0572
λ_5		0.4839					
λ_6			0.4615				
λ_7				0.4468			
λ_8					0.4364	0.1291	0.1143
λ_9						0.2903	
λ_{10}							0.2857
ρ_6	1	0.7742	0.4615	0.2553	0.1091	0.0968	
ρ_7		0.2258	0.5385	0.7447	0.8909	0.9032	1
$ E_{\mathcal{P}} $	24	31	39	47	55	62	70
$(\frac{E_b}{N_0})^*$	0.8172	0.7463	0.6888	0.6305	0.6011	0.4964	0.4813

TABLEAU 5.8 Paires de distributions adaptées aux codes convolutionnels récurrents doublement-orthogonaux de taux de codage $r = b/c$ égal à $2/3$, $b \in \{4, 5, 6, 7, 8, 9, 10\}$, offrant de bons seuils de convergence pour le canal AWGN. Pour chaque paire de distribution, le tableau indique la valeur du seuil $(\frac{E_b}{N_0})^*$ en decibel ainsi que le nombre d'arêtes $|E_{\mathcal{P}}|$ constituant le protographe.

$\lambda_i \mid \rho_i$	Dimensions de $\mathbf{H}^T(D) : c \times (c - b)$						
	12×4	15×5	18×6	21×7	24×8	27×9	30×10
λ_2	0.2162	0.2667	0.2456	0.1944	0.1905	0.1875	0.1739
λ_3	0.2433	0.4000	0.3684	0.4167	0.4286	0.4375	0.3913
λ_4	0.5405		0.0702				
λ_5		0.3333					
λ_6			0.3158				
λ_7				0.3889			
λ_8					0.3809		
λ_9						0.3750	
λ_{10}							0.4348
ρ_9	0.7297	1	0.4737				
ρ_{10}	0.2703		0.5263	0.6944	0.4762	0.3125	
ρ_{11}				0.3056	0.5238	0.6875	0.4783
ρ_{12}							0.5217
$ E_{\mathcal{P}} $	37	45	57	72	84	96	115
$(\frac{E_b}{N_0})^*$	1.6143	1.5256	1.4987	1.4534	1.4076	1.3614	1.3427

5.2.2 Résultats de simulations des codes RCDO pour le canal binaire AWGN non quantifié

Dans cette sous-section, nous présentons les résultats de simulations par ordinateur des codes convolutionnels récurrents doublement orthogonaux de taux de codage $r = 1/4, 1/3, 1/2$ et $2/3$ pour le canal AWGN. Le décodeur itératif utilisé est l'algorithme à passage de messages décrit au chapitre 3. Notons que toutes les matrices de contrôle des codes RCDO présentées dans cette section se trouvent à l'Annexe C.

Dans un premier temps, nous présentons à la Figure 5.8 les performances d'erreur d'un code RCDO de taux de codage $1/2$ et de mémoire m_s égal à 62. Ce code possède une matrice de contrôle $\mathbf{H}^T(D)$ de dimensions 16×8 et un graphe biparti de Tanner dont les noeuds variables et les noeuds de contraintes sont tous de degré constant et égal à 3 et 6 respectivement. Comme nous l'avons présenté au Tableau 5.5, les codes convolutionnels RCDO réguliers définis par la paire de distribution (x^2, x^5) peuvent au mieux s'approcher à 0.92 dB de la limite de Shannon, car l'algorithme itératif de décodage converge théoriquement à 1.11 dB. Comme nous pouvons le voir sur cette figure, après avoir effectué 25 itérations, ce code offre des performances d'erreur qui se situent à environ 0.5 dB de la valeur du seuil correspondant au graphe biparti. À titre de comparaison, nous avons ajouté sur cette figure les résultats obtenus par (MacKay, 1999) d'un code en blocs LDPC régulier de taux de codage $1/2$ possédant une longueur de bloc n égale à 1008 et un graphe biparti de Tanner possédant la paire de distributions (x^2, x^5) . Nous pouvons remarquer sur cette figure que le code RCDO offre un gain de codage supplémentaire d'environ 1 dB par rapport au code LDPC en blocs de Mackay.

Nous présentons aussi sur cette même figure deux autres codes convolutionnels récurrents de taux de codage $r = 1/2$, mais pour lesquels les éléments $\alpha_{m,n}$ de la matrice $\mathbf{H}^T(D)$ ne répondent pas complètement aux conditions de la double orthogonalité. Le premier code répond uniquement à la première condition de la définition de la double orthogonalité tandis que le second code ne répond qu'aux deux premières conditions de la définition de la double orthogonalité. Il en résulte que le premier code possède des cycles de longueur 6 dans son graphe biparti et le second code contient des cycles de longueur 8. Ces deux codes possèdent eux aussi comme le code RCDO une mémoire égal à 62. De cette façon, tous les codes convolutionnels nécessitent la même complexité de décodage ainsi que la même latence. On peut donc remarquer qu'il

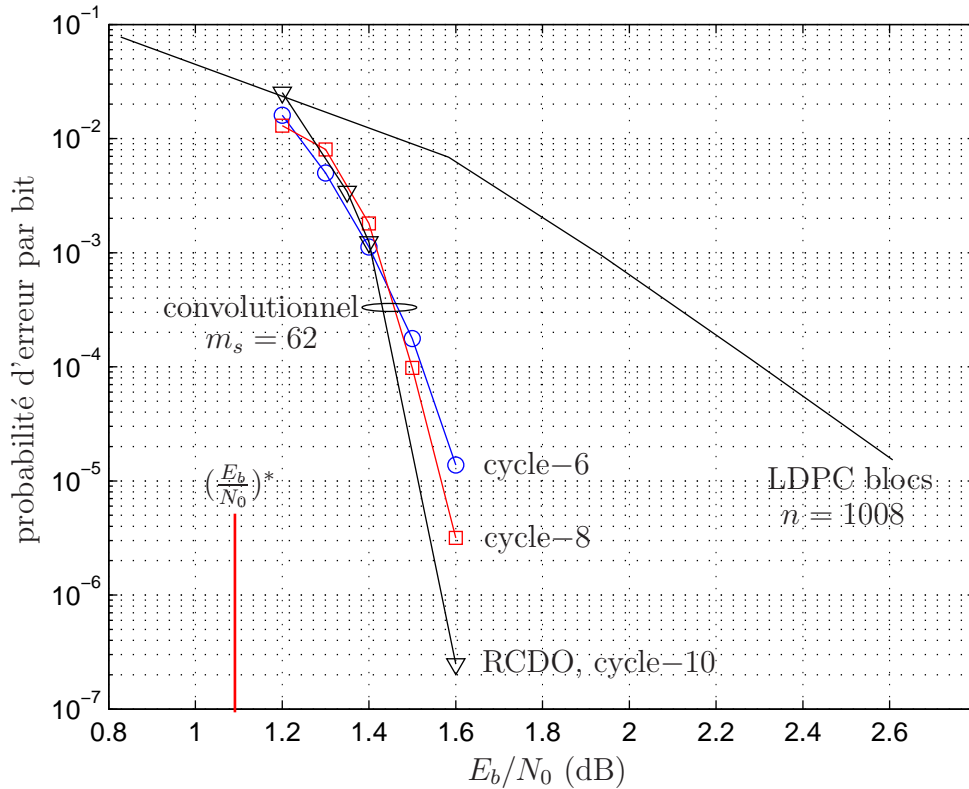


FIGURE 5.8 Comparaison entre les performances d'erreur de codes convolutionnels récurrents de taux de codage $r = 8/16$ avec celles d'un code bloc LDPC (MacKay, 1999) où $n = 1008$. Pour chaque codes, le graphe biparti possède la paire de distribution (x^2, x^5) .

est possible d'obtenir un gain en probabilité d'erreur équivalent à environ un ordre de grandeur pour chacune des conditions de la double orthogonalité vérifiées par le code convolutionnel récurrent. D'où l'avantage des conditions de la double orthogonalité pour générer un code convolutionnel récurrent LDPC.

À la Figure 5.9, nous présentons les performances d'erreur obtenues par un autre code RCDO régulier de taux de codage $1/2$ possédant la paire de distribution (x^2, x^5) . La mémoire m_s de ce code est égale à 149 et la matrice de contrôle est de dimension (30×15) . D'une part, nous pouvons remarquer que les performances d'erreur s'améliorent d'une itération à l'autre et qu'elles s'approchent de la valeur du seuil pour cette paire de distributions. En fait, après 100 itérations, les performances d'erreur se situent à moins de 0.1 dB de la valeur du seuil théorique et offrent un gain

de codage additionnel par rapport au code RCDO de la Figure 5.8. En revanche, les quelques dixièmes de décibel gagnés par ce code se font au détriment d'une augmentation substantielle de la complexité et de la latence associée au décodage lorsqu'on le compare avec le code RCDO de mémoire 62. Effectivement, le code RCDO pour lequel la valeur m_s est égale à 149 possède un protographe ayant 90 arêtes tandis que le protographe correspondant au code RCDO avec m_s égal à 62 contient 48 arêtes. Il s'ensuit que l'implémentation d'une itération de décodage pour le code RCDO ayant $m_s = 149$ nécessite environ 1.87 fois plus d'opérateurs \mathcal{R} et d'additions de nombres réels que l'implémentation d'une itération pour le code RCDO ayant $m_s = 62$.

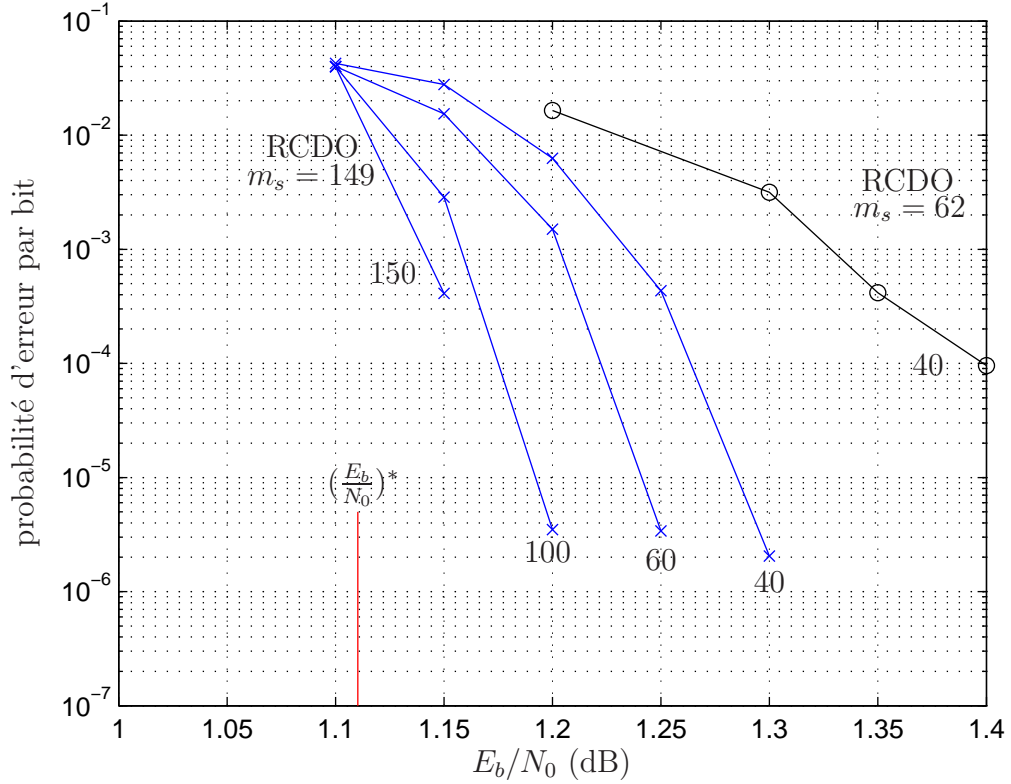


FIGURE 5.9 Performances d'erreur de codes RCDO réguliers (3,6) de taux de codage 15/30 pour plusieurs itérations, $m_s = 149$.

À la Figure 5.10, nous présentons quatre codes RCDO irréguliers de taux de codage $r = 1/4$, $1/3$, $1/2$ et $2/3$. Notons que tous les codes RCDO présentés sur cette figure, à l'exception du code RCDO de taux de codage $1/4$, possèdent un protographe

dont le degré maximum des noeuds variables est égal à 10 et possèdent les paires de distributions indiquées dans les Tableaux 5.6, 5.7 et 5.8. Dans le but d’alléger le texte, les matrices de contrôle de ces quatre codes RCDO se trouvent à l’Annexe C.

Sur cette figure, nous pouvons remarquer que les codes RCDO, dont le proto-graphe est irrégulier, peuvent approcher la limite de Shannon. En comparaison avec le meilleur code S-CDO (Roy *et al.*, 2007), les codes RCDO offrent un gain de codage asymptotique supplémentaire d’au moins 1 dB. De plus, moins de 2.2 dB séparent les performances d’erreur du code RCDO, $r = 1/4$, de la limite absolue de Shannon. D’autre part, nous pouvons aussi remarquer que le coût associé à l’amélioration des performances d’erreur se traduit par une augmentation de la largeur de bande nécessaire pour transmettre les signaux, ainsi que par une augmentation de la complexité lors du décodage, ainsi que par une augmentation de latence lors du décodage. En comparaison avec le cas sans codage, pour une probabilité d’erreur fixé à 10^{-5} , le code RCDO ayant $r = 1/4$ permet la transmission des signaux avec huit fois moins de puissance, c’est-à-dire un gain de codage d’environ 9 dB.

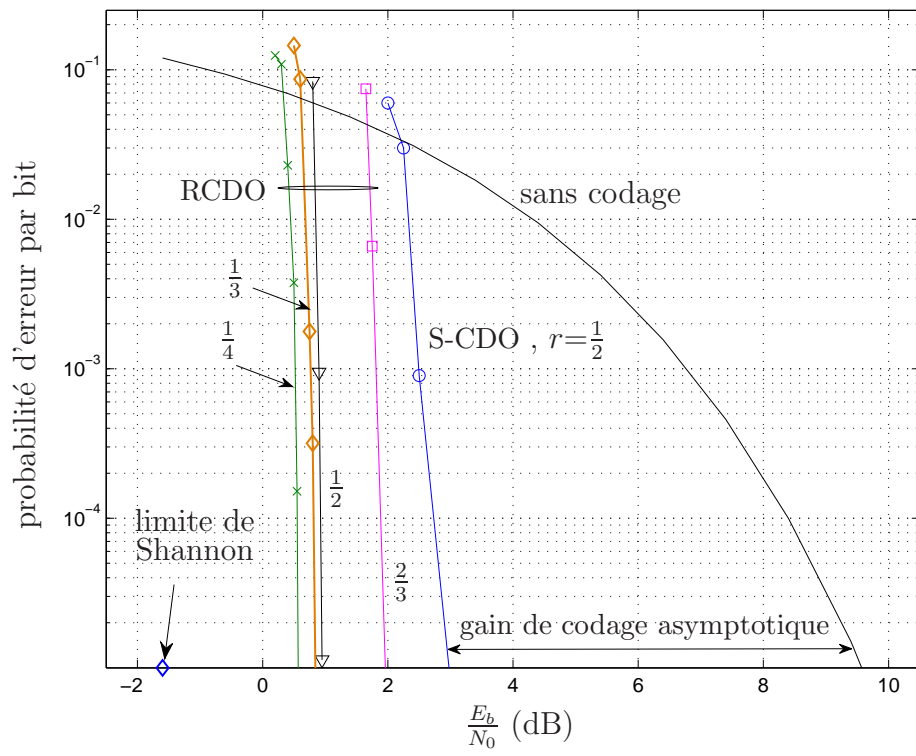


FIGURE 5.10 Performances d'erreur de codes convolutionnels RCDO irréguliers de taux de codage $r = 1/4$, $1/3$, $1/2$ et $2/3$.

5.3 Performances d’erreur associées aux codes RCDO réguliers (3,6) doublement orthogonaux de faible complexités

Suite aux sections précédentes, nous avons entrepris la recherche de codes convolutionnels RCDO à l’aide d’une heuristique permettant la minimisation de la latence introduite par une itération de décodage, c’est-à-dire que nous avons minimisé la valeur m_s , qui représente la longueur du plus grand registre à décalage qui compose le codeur convolutionnel. Nous présentons à l’Annexe B le pseudo-code qui représente l’heuristique de recherche qui fût développée. Cet algorithme, a pour but de minimiser la valeur m_s pour une matrice de contrôle $\mathbf{H}^T(D)$ de dimensions fixes et qui répond à une certaine paire de distributions (λ, ρ) .

De cette recherche par ordinateurs, nous présentons sept codes doublement orthogonaux pour lesquels nous résumons sous forme de tableau les différents paramètres liés à la complexité de calculs du codeur et du décodeur. Dans un premier temps, le Tableau 5.9 résume les différentes valeurs obtenues suite à la minimisation du paramètre m_s en fonction du nombre de registres à décalage en parallèle. Comme nous pouvons le constater, la taille du plus grand registre à décalage tend à croître avec la taille de la matrice de contrôle. Ceci s’explique par le fait qu’il y a plus de connexions dans une matrice de contrôle plus grande. Par exemple, le code 1 possède un protographe ayant 24 arêtes et le code 7 possède un protographe ayant 60 arêtes. Il s’en suit qu’il y a plus de connexions qui doivent vérifier la double orthogonalité pour le code 7 qu’il y en a pour le code 1 ce qui fait croître la valeur m_s . Le Tableau 5.10 présente le nombre d’additions modulo 2 ainsi que la mémoire totale nécessaire à la réalisation des encodeurs correspondant aux codes RCDO.

Au Tableau 5.11 nous résumons les différents paramètres liés à la réalisation d’une itération de décodage pour les sept codes RCDO réguliers présentés à l’Annexe—C. Comme nous pouvons le constater, plus le nombre d’arêtes composant le protographe est élevé plus la complexité de la réalisation matérielle d’une itération de décodage est élevée.

À la Figure 5.11, nous présentons les performances d’erreur des sept codes RCDO réguliers (3, 6) (Roy *et al.*, 2012). Pour cette paire de distribution, le seuil de convergence asymptotique est égal à 1.11 dB. Comme nous pouvons le remarquer, le code

TABLEAU 5.9 Codes convolutionnels LDPC réguliers (3,6) doublement orthogonaux introduisant une faible latence

code RCDO	Nombre de registres à décalage ($c - b$)	Taille du plus grand registre m_s
code 1	4	33
code 2	5	33
code 3	6	34
code 4	7	34
code 5	8	40
code 6	9	43
code 7	10	49

TABLEAU 5.10 Complexité de calculs et mémoires totales nécessaires pour la réalisation des codeurs convolutionnels associés aux codes du Tableau 5.9

code RCDO	Nombre d'additions modulo 2 $ E_{\mathcal{P}} - 2(c - b)$	mémoire totale m_{tot}
code 1	16	113
code 2	20	143
code 3	24	168
code 4	28	200
code 5	32	262
code 6	36	356
code 7	40	417

TABLEAU 5.11 Complexité associée à la réalisation matérielle d'une itération de décodage des codes RCDO du Tableau 5.9.

code RCDO	nombre d'arêtes $ E_{\mathcal{P}} $	délai de décodage $m_s + 1$	unités de mémoire $(E_{\mathcal{P}} + c)(m_s + 1)$	nombre d'opérateurs \mathcal{R} $(c - b)\bar{d}_\rho$	nombre d'additions $ E_{\mathcal{P}} \lambda'(1)$
code 1	24	34	1088	24	48
code 2	30	34	1360	30	60
code 3	36	35	1680	36	72
code 4	42	35	1960	42	84
code 5	48	41	2624	48	96
code 6	54	44	3168	54	108
code 7	60	50	3999	60	120

RCDO 7 est celui qui offre les meilleures performances d'erreur parmi l'ensemble des codes RCDO présentés. Par exemple, pour une probabilité d'erreur fixée à 10^{-5} , le code 7 offre un gain de codage asymptotique additionnel d'environ 0.5 dB par rapport au code 1. De plus, nous avons aussi indiqué sur cette figure les performances d'erreur d'un code en blocs LDPC régulier (3, 6) de taux de codage 1/2. La longueur du bloc n pour ce code LDPC est égale à 1008. Comme nous pouvons le constater tous les codes RCDO offrent des performances d'erreur supérieures à celles du code en blocs LDPC, et ce, en effectuant deux fois moins d'itérations. Du point de vue de la complexité d'implémentation, le code LDPC en blocs nécessite un décodeur qui utilise 3024 opérateurs \mathcal{R} tandis que le décodeur associé au code 7 RCDO nécessite au total 1500 opérateurs \mathcal{R} . Les codes RCDO présentés sur cette figure permettent donc de diminuer substantiellement le nombre d'opérateurs nécessaires à la mise à jour des messages extrinsèques. Finalement, mentionnons que tous les codes RCDO réguliers (3,6) offrent de bien meilleures performances d'erreurs que celles offertes par les codes non rékursifs CDO et S-CDO.

À la Figure 5.11, nous avons reproduit les performances d'erreur, après 100 itérations de décodage, d'un code régulier (3,6) LDPC-C de taux de codage 8/16 généré selon la technique présentée dans (Pusane *et al.*, 2011) et qui possède une longueur de contrainte $c(m_s + 1)$ égale à 2048. Comme nous pouvons le constater, le code LDPC-C offre un gain de codage supplémentaire en comparaison avec les codes (3,6) RCDO

optimisés. Ceci s'explique par le fait que la longueur de contrainte $c(m_s + 1)$ du code LDPC-C, qui a été généré de façon aléatoire, est plus grande que celle des sept (7) codes RCDO. Notons qu'à l'équation (5.18) nous présentons la matrice de parité qui génère un code RCDO de taux de codage 8/16 ayant $m_s = 127$. Les performances d'erreur de ce code sont présentées à la Figure 5.11 et elles sont notées RCDO 8. Comme nous pouvons le constater, le code RCDO 8 permet d'atteindre, pour un même nombre d'itération, de meilleures performances d'erreur que tous les autres codes RCDO présentés sur cette Figure. Ceci s'explique, car la valeur $c(m_s + 1)$ de ce code est plus élevée que celle des codes RCDO 1 à 7. Cependant, lorsque nous effectuons un plus grand nombre d'itérations, nous pouvons remarquer que le code RCDO 8 peut offrir des performances d'erreur pratiquement égales à celles du code (3,6) LDPC-C pour lequel 100 itérations ont été effectuées. Deux constats peuvent donc être effectués à partir de ces résultats. Premièrement, pour une valeur de contrainte égale à un code LDPC-C généré de façon aléatoire, il est possible avec un code RCDO d'effectuer un moins grand nombre d'itérations qu'il n'en faut pour atteindre les mêmes performances d'erreur qu'avec un code LDPC-C généré de façon aléatoire. Ceci se traduit donc en une réduction de l'effort de calcul ainsi qu'en une réduction du nombre d'unités de mémoire requises pour obtenir des performances d'erreur similaires à celles du code LDPC-C. Deuxièmement, le code RCDO permet aussi de réduire la latence lors du décodage, car il nécessite deux fois moins d'itérations que le code LDPC-C.

$$\mathbf{H}^T(D) = \begin{pmatrix} 0 & 0 & D^{121} & D^{96} & 0 & D^{21} & 0 & 0 \\ 0 & 0 & D^{20} & D^{17} & 0 & D^{83} & 0 & 0 \\ 0 & 0 & 0 & D^{51} & D^{117} & 0 & 0 & D^{45} \\ D^{127} & D^{78} & 0 & 0 & D^{77} & 0 & 0 & 0 \\ 0 & 0 & 0 & D^{110} & 0 & D^3 & 0 & D^{90} \\ D^{103} & 0 & D^{122} & 0 & D^{101} & 0 & 0 & 0 \\ D^{114} & 0 & D^{118} & 0 & D^{86} & 0 & 0 & 0 \\ 0 & D^{11} & D^{58} & 0 & 0 & D^{33} & 0 & 0 \\ 1 & 0 & 0 & 0 & D^{75} & 0 & D^{12} & 0 \\ 0 & 1 & 0 & D^{94} & 0 & 0 & D^{99} & 0 \\ D^{15} & 0 & 1 & 0 & 0 & D^5 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & D^{82} & D^{93} \\ D^{115} & 0 & 0 & 0 & 1 & 0 & 0 & D^{57} \\ 0 & D^{35} & 0 & 0 & 0 & 1 & D^{116} & 0 \\ 0 & D^{68} & 0 & 0 & 0 & 0 & 1 & D^{84} \\ 0 & D^{120} & 0 & 0 & 0 & 0 & D^4 & 1 \end{pmatrix} \quad (5.18)$$

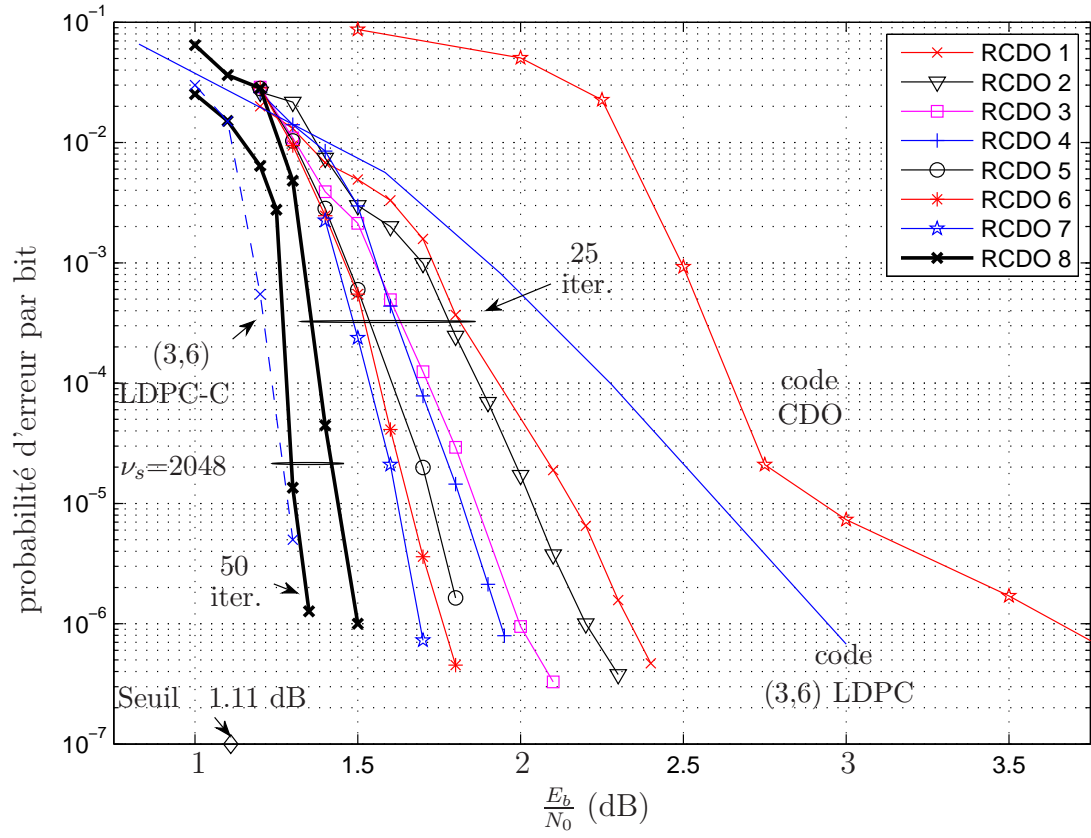


FIGURE 5.11 Performances d'erreur pour les codes réguliers (3,6) RCDO de taux de codage 1/2 présentés au Tableau 5.9. Aussi reproduit, les performances d'erreur d'un code en bloc régulier (3,6) LDPC de taux de codage 1/2 ainsi que celles d'un code CDO ayant $m_s=1698$ et $r = 1/2$.

5.4 Conclusion

Dans cette section, nous avons étudié plus précisément l'importance du choix des distributions sur le degré des noeuds des protographes associées aux codes convolutionnels RCDO. D'une part, nous avons montré qu'il existe des protographes irréguliers adaptés aux codes convolutionnels RCDO qui possèdent un seuil de convergence asymptotique qui s'approche de la limite de Shannon. Nous avons montré que ceci était vrai pour les canaux symétriques et sans mémoire BEC et AWGN. Nous avons fourni pour ces deux canaux les meilleures paires de distributions adaptées aux taux de codage $1/3$, $1/2$ et $2/3$, sous les contraintes que nous avons présenté. Les Tableaux 5.1, 5.2, 5.3, 5.6, 5.7 et 5.8 synthétisent ces résultats. Pour le canal BEC, la paire de distributions adaptée aux codes RCDO de taux de codage $1/2$ s'approche au mieux à 10.12% de la limite de Shannon. Pour le canal binaire AWGN, la paire de distributions qui s'approche le plus près de la limite de Shannon possède un seuil théorique qui est à environ 0.3 dB. Comme nous l'avons montré à la Figure 5.6 le seuil théorique de convergence de l'algorithme itératif des codes RCDO est un peu plus élevé que celui des codes en blocs LDPC. C'est-à-dire que théoriquement les codes en blocs LDPC peuvent fournir de meilleures performances d'erreur à faible rapport signal sur bruit. Cependant, l'écart entre les seuils théoriques des codes RCDO et LDPC est très faible, et s'explique par le fait que les paires de distributions adaptées aux codes LDPC le sont pour des codes LDPC ayant des longueurs de blocs très élevées ce qui n'est pas le cas pour les codes RCDO présentés. À partir de notre analyse sur la complexité provenant du chapitre précédent, nous avons pu montrer que la complexité de calculs associée à une itération croît très rapidement au fur et à mesure où les seuils théoriques de convergence des codes RCDO s'approchent de la limite de Shannon.

En plus d'avoir présenté les paires de distributions adaptées aux codes RCDO, nous avons aussi trouvé plusieurs codes RCDO. Toutes les matrices de contrôle des codes RCDO adaptés aux canaux BEC et AWGN présentés dans ce chapitre se retrouvent à l'Annexe—C. Nos résultats de simulations nous montrent que les conditions de double orthogonalité imposées aux connexions des codeurs RCDO permettent l'amélioration de la convergence du décodeur itératif, et ainsi améliorent les performances d'erreur, lorsque nous les comparons avec des codes convolutionnels pour lesquels les connexions ne respectent pas toutes les conditions de la double orthogonalité. Ceci s'explique par le fait que la double orthogonalité augmente la taille du plus petit

cycle dans le graphe de Tanner des codes convolutionnels récurrents multiregistres ce qui améliore l'estimation des symboles d'une itération à l'autre. De plus, nous avons montré qu'il est possible d'approcher la limite asymptotique du seuil de convergence théorique associé aux codes RCDO ayant un protographe régulier (3,6). D'ailleurs, nous avons aussi présenté plusieurs codes RCDO réguliers (3,6) pour lesquels nous avons minimisé la mémoire du plus grand registre à décalage, afin de réduire la latence introduite par le décodeur itératif. Comme nous l'avons montré pour ces codes RCDO réguliers, les performances d'erreur s'améliorent plus le codeur RCDO possède de registres à décalage en parallèle. Cependant, ce gain en performances d'erreur se traduit en une augmentation de la complexité de calculs lors des itérations effectuées par le décodeur itératif. Les codes RCDO réguliers (3,6) possèdent des performances d'erreur largement supérieures à celles de tous les codes CDO présentés auparavant. Un gain de codage additionnel de plus de 1.5 dB a pu être observé en comparaison avec les meilleurs codes CDO existants. Nous avons aussi montré que leurs performances d'erreur étaient concurrentes à celles des codes réguliers (3,6) LDPC-C construits de façon aléatoire lorsque la longueur de contrainte entre les codes RCDO et LDPC-C était identique. De plus, nos simulations indiquent qu'il est possible d'atteindre ces performances d'erreur en effectuant deux fois moins d'itérations qu'il n'en faut pour les codes LDPC-C. Ceci se traduit en un avantage en faveur des codes RCDO étant donné que la latence de décodage est plus faible que celle introduite par les codes réguliers LDPC-C ayant la même longueur de contrainte.

Nous avons aussi montré qu'il est possible d'améliorer les performances d'erreur des codes RCDO réguliers en utilisant des paires de distributions irrégulières (λ, ρ) . Comme nous l'avons montré, un gain de codage supplémentaire d'environ 0.6 dB peut être obtenu en utilisant un code RCDO irrégulier plutôt qu'un code RCDO régulier (3,6). Toutefois, ce gain est obtenu au prix d'une plus grande complexité d'implémentation que celle nécessaire pour les codes RCDO réguliers.

Chapitre 6

Codes RCDO imbriqués

Dans ce chapitre, nous présentons les codes RCDO imbriqués qui permettent d'atteindre plusieurs taux de codage à la fois à partir d'un seul code RCDO mère. Cette propriété est attrayante, car certains systèmes de communications en tirent avantage. Par exemple, les codes RCDO imbriqués pourraient être utilisés dans des systèmes de communications sans fil, pour lesquels le canal de transmission sans fil varie dans le temps. En effet, lorsque le canal est non fiable, il devient possible de diminuer le taux de codage du codeur RCDO mère de façon à mieux protéger les symboles transmis. À l'inverse, lorsque le canal est très fiable, et perturbe peu les signaux transmis, l'émetteur peut changer le taux de codage du codeur RCDO pour un taux de codage plus élevé. Ce chapitre introduit donc les définitions, les concepts et les problématiques reliés aux design de codes RCDO mères. De plus, nous présentons certains résultats préliminaires qui montrent qu'il est possible de générer des codes RCDO mères qui offrent des gains de codage substantiellement plus élevés que ceux obtenus en utilisant les meilleurs codes perforés convolutionnels non récursifs doublement orthogonaux présentés dans (Haccoun et Cardinal, 2005) ainsi que leur version simplifiée présentée dans (Roy *et al.*, 2007). Notons, qu'à ce jour, la technique présentée dans ce chapitre est, à notre connaissance, la seule technique de codage permettant d'obtenir des codes LDPC convolutionnels pouvant atteindre plusieurs taux de codage simultanément, et ce, sans utiliser un mécanisme de perforation.

6.1 Introduction aux codes convolutionnels RCDO imbriqués

Les codes RCDO imbriqués sont simplement des codes RCDO qui s'obtiennent en changeant le nombre de symboles acceptés à l'entrée d'un codeur RCDO mère de taux de codage $r = b/c$, $b < c$, $b > 1$. La Figure 6.1 présente l'exemple d'un codeur RCDO mère de taux de codage 3/6, où la première entrée n'accepte aucun symbole

d'information provenant de la source. Il s'en suit que le codeur RCDO mère ne génère aucun symbole à la sortie correspondant à l'entrée bloquée, et le taux de codage résultant devient égal à $2/5$. Cette opération vient donc modifier le taux de codage du codeur RCDO mère et peut être facilement réaliser en utilisant un démultiplexeur en entrée ainsi qu'un multiplexeur à la sortie du codeur RCDO.

En généralisant à un codeur RCDO de taux de codage égal à b/c , on peut donc atteindre les taux de codage $(\frac{b-k}{c-k})$, $1 \leq k < b$, lorsque k entrées d'un codeur RCDO mère n'admettent aucun symbole. Nous pouvons donc remarquer que le nouveau taux de codage généré par la méthode proposée est plus faible que celui du code RCDO mère. Notons que tous les codes RCDO présentés dans les chapitres antérieurs sont tous des codes RCDO mères. Toutefois, comme nous le montrons à la prochaine section, certains critères supplémentaires doivent être pris en compte lors de la construction de codes RCDO mères si nous désirons atteindre de bonnes performances d'erreur pour tous les taux de codage que nous désirons atteindre avec le codeur RCDO mère.

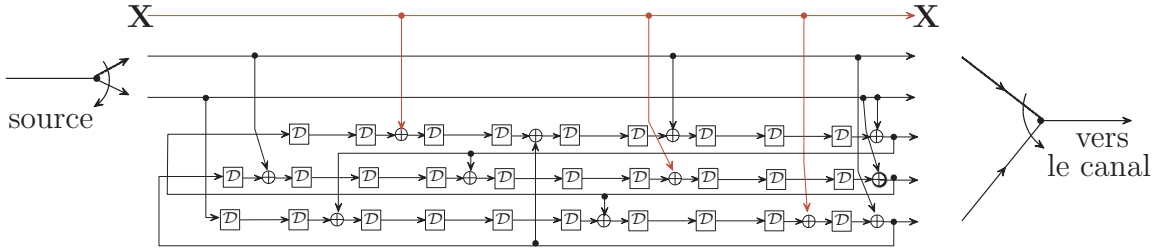


FIGURE 6.1 Exemple d'un codeur convolutionnel récursif composé de 3 registres en parallèle et d'un taux de codage $3/6$ avec lequel il est possible d'atteindre le taux de codage $2/5$.

6.2 Protographes et matrices de contrôle des codes RCDO imbriqués

Comme nous l'avons présenté dans cette thèse, nous pouvons associer des protographes aux codes convolutionnels récursifs RCDO. Ces protographes sont définis entre autres, par leur paire de distributions (λ, ρ) , à partir de laquelle il est pos-

sible d'évaluer le seuil de convergence asymptotique du décodeur itératif. Le chapitre précédent nous a permis d'observer que de toutes les paires de distributions RCDO, certaines d'entrent elles offrent de meilleurs seuils de convergence asymptotiques que d'autres paires de distributions RCDO. De cette observation, nous pouvons donc nous demander quelle est l'impact sur la paire de distributions des codes RCDO mères lorsque nous réduisons le nombre d'entrées aux codeurs RCDO mères. Pour comprendre les effets associés à la réduction du taux de codage d'un code mère, nous présentons dans cette section les matrices de contrôle $\tilde{\mathbf{H}}_k^T(D)$ ainsi que les proto-graphes $\tilde{G}_{\mathcal{P}_k}$ qui correspondent aux codes RCDO imbriqués de taux de codage $(\frac{b-k}{c-k})$ qui sont obtenus en bloquant k entrées (et sorties) des codeurs RCDO mères de taux de codage b/c .

Au Chapitre 3, nous avons montré que les équations de contraintes associées à un code RCDO de taux de codage b/c sont représentés par l'équation (6.1).

$$c_i^{(n)} = v_i^{(b+n)} + \sum_{\substack{m=1 \\ m \neq b+n}}^c v_{i-\alpha_{m,n}}^{(m)}, \quad 1 \leq n \leq c-b \quad (6.1)$$

Ces équations de contraintes nous permettent de définir la matrice de contrôle $\mathbf{H}^T(D)$ qui est associée aux codes RCDO. Cette matrice de contrôle, de dimension $c \times (c-b)$, est représentée par l'équation (6.2).

$$\mathbf{H}^T(D) = \begin{pmatrix} h_{1,1}D^{\alpha_{1,1}} & \dots & h_{1,(c-b)}D^{\alpha_{1,(c-b)}} \\ h_{2,1}D^{\alpha_{2,1}} & \dots & h_{2,(c-b)}D^{\alpha_{2,(c-b)}} \\ \vdots & & \vdots \\ h_{b,1}D^{\alpha_{b,1}} & \dots & h_{b,(c-b)}D^{\alpha_{b,(c-b)}} \\ h_{b+1,1}D^{\alpha_{b+1,1}} & \dots & h_{b+1,(c-b)}D^{\alpha_{b+1,(c-b)}} \\ \vdots & & \vdots \\ h_{c,1}D^{\alpha_{c,1}} & \dots & h_{c,(c-b)}D^{\alpha_{c,(c-b)}} \end{pmatrix} \quad (6.2)$$

Notons qu'en n'admettant aucun symbole sur k entrées d'un codeur RCDO mère a pour effet d'éliminer certains termes de la somme $\sum_{\substack{m=1 \\ m \neq b+n}}^c v_{i-\alpha_{m,n}}^{(m)}$ dans l'équation (6.1). Par exemple, si la première entrée(sortie) d'un codeur est bloquée, alors les termes $v_i^{(1)}$ ne peuvent plus faire partie de la somme $\sum_{\substack{m=1 \\ m \neq b+n}}^c v_{i-\alpha_{m,n}}^{(m)}$, car le démultiplexeur en entrée du codeur RCDO ne fournit plus de symbole d'information provenant de

la source sur cette entrée du codeur RCDO. Ceci fait en sorte que la matrice de contrôle $\tilde{\mathbf{H}}_k^T(D)$ d'un code RCDO imbriqué de taux de codage $(\frac{b-k}{c-k})$ est de dimension $(c-k) \times (c-b)$ et est représentée par l'équation (6.3). Cette matrice s'obtient facilement à partir de la matrice de contrôle du code RCDO mère $\mathbf{H}^T(D)$. En effet, $\tilde{\mathbf{H}}_k^T(D)$ s'obtient en éliminant les k lignes de $\mathbf{H}^T(D)$ qui correspondent aux k entrées bloquées du codeur RCDO mère. Par exemple, si la seconde entrée d'un codeur RCDO mère est bloquée, alors la matrice de contrôle correspondant au code RCDO résultant $\tilde{\mathbf{H}}_1^T(D)$ s'obtient en éliminant la seconde ligne de la matrice de contrôle $\mathbf{H}^T(D)$ du code RCDO mère. La matrice $\tilde{\mathbf{H}}_k^T(D)$ est donc imbriquée dans la matrice $\mathbf{H}^T(D)$.

$$\tilde{\mathbf{H}}_k^T(D) = \begin{pmatrix} h_{1,1}D^{\alpha_{1,1}} & \dots & h_{1,(c-b)}D^{\alpha_{1,(c-b)}} \\ h_{2,1}D^{\alpha_{2,1}} & \dots & h_{2,(c-b)}D^{\alpha_{2,(c-b)}} \\ \vdots & & \vdots \\ h_{b-k,1}D^{\alpha_{b-k,1}} & \dots & h_{b-k,(c-b)}D^{\alpha_{b-k,(c-b)}} \\ h_{b-k+1,1}D^{\alpha_{b-k+1,1}} & \dots & h_{b-k+1,(c-b)}D^{\alpha_{b-k+1,(c-b)}} \\ \vdots & & \vdots \\ h_{c-k,1}D^{\alpha_{c-k,1}} & \dots & h_{c-k,(c-b)}D^{\alpha_{c-k,(c-b)}} \end{pmatrix} \quad (6.3)$$

Comme nous l'avons présenté au Chapitre 3, nous pouvons faire correspondre un protographe $G_{\mathcal{P}} = (V_v \cup V_c, E)$ aux codes RCDO possédant une matrice de contrôle $\mathbf{H}^T(D)$, et vice versa. Chaque ligne de la matrice de contrôle représente un noeud de l'ensemble V_v et chaque colonne représente un noeud de contraintes de l'ensemble V_c . Un noeud $v^{(m)}$ de V_v est relié au noeud $c^{(n)}$ de V_c par une arête $e_{m,n}$ si la valeur $h_{m,n}$ de $\mathbf{H}^T(D)$ est non nul. Le protographe $G_{\mathcal{P}}$ à gauche de la Figure 6.2 représente le protographe général associé à un code RCDO de taux de codage b/c . Le protographe $\tilde{G}_{\mathcal{P}_k}$ correspondant à un code RCDO imbriqué de taux de codage $(\frac{b-k}{c-k})$ représente simplement le sous-graphe induit $\tilde{G}_{\mathcal{P}_k} = G_{\mathcal{P}} \setminus \{\bigcup_k v^{(k)} : k \text{ indice du symbole bloqué}\}$ obtenu à partir du protographe $G_{\mathcal{P}}$ du code RCDO mère. Par exemple, à la droite de la Figure 6.2, nous présentons le protographe $\tilde{G}_{\mathcal{P}(b-1)}$ correspondant au code RCDO imbriqué de taux de codage $\frac{1}{c-b+1}$ obtenu à partir du protographe à la gauche sur la Figure 6.2.

Étant donné que le protographe $\tilde{G}_{\mathcal{P}_k}$ est obtenu à partir du protographe $G_{\mathcal{P}}$ associé au code RCDO mère, nous pouvons donc dire en quelque sorte que la paire de distributions $(\tilde{\lambda}, \tilde{\rho})_k$ associée au protographe $\tilde{G}_{\mathcal{P}_k}$ est imbriquée dans la paire de

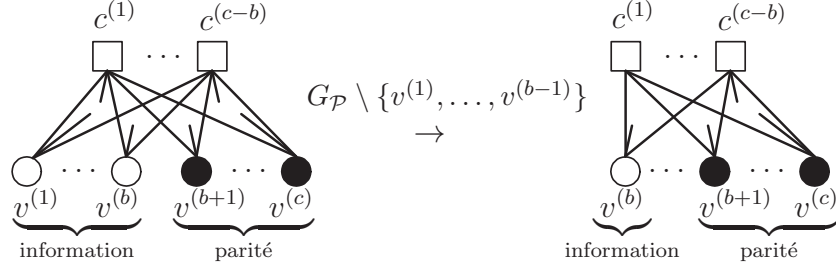


FIGURE 6.2 À gauche, le protographe général $G_{\mathcal{P}}$ d'un ensemble de codes RCDO. À droite, le protographe imbriqué $\tilde{G}_{\mathcal{P}(b-1)}$ qui correspond au sous-graphe induit $G_{\mathcal{P}} \setminus \{v^{(1)}, \dots, v^{(b-1)}\}$.

distributions (λ, ρ) du protographe $G_{\mathcal{P}}$. Nous observons donc que la diminution du taux de codage du code mère vient modifier la paire de distributions (λ, ρ) , et implicitement vient aussi influencer les propriétés de convergence associées du décodeur itératif. Par conséquent, il s'avère important d'effectuer judicieusement le design du protographe des codes RCDO mères si l'on désire obtenir des paires de distributions imbriquées $(\tilde{\lambda}, \tilde{\rho})_k$ qui offrent de bons seuils asymptotiques de convergence pour tous les taux de codage que l'on désire atteindre à partir du protographe du code RCDO mère.

Par exemple, nous avons pu observer à partir des résultats présentés aux Tableaux 5.7 et 5.8, que la paire de distributions $(0.2162x + 0.2433x^2 + 0.5405x^3, 0.7297x^8 + 0.2703x^9)$ du Tableau 5.8, qui est adaptée aux codes RCDO de taux de codage 8/12, peut être utilisée pour générer la paire de distributions $(0.3333x + 0.6667x^3, x^5)$ du Tableau 5.7, qui est adaptée aux codes RCDO de taux de codage 4/8. Notons que ces deux paires de distributions représentent les paires de distributions qui offrent les meilleurs seuils de convergence théorique lorsque le degré maximum d'un noeud variable dans le protographe d'un code RCDO est fixé à 4. Le protographe $G_{\mathcal{P}}$ de la Figure 6.3 est composé de 12 noeuds variables, de 4 noeuds de contraintes ainsi que de 37 arêtes, ce protographe $G_{\mathcal{P}}$ possède la paire de distributions $(0.2162x + 0.2433x^2 + 0.5405x^3, 0.7297x^8 + 0.2703x^9)$. Pour obtenir la paire de distributions $(0.3333x + 0.6667x^3, x^5)$ à partir $G_{\mathcal{P}}$, il suffit simplement d'éliminer du protographe $G_{\mathcal{P}}$ de la Figure 6.3 les 4 noeuds variables $v^{(1)}, \dots, v^{(4)}$ ainsi que les 13 arêtes incidentes aux noeuds variables éliminés. Ainsi le protographe $\tilde{G}_{\mathcal{P}_4} = G_{\mathcal{P}} \setminus \{v^{(1)}, \dots, v^{(4)}\}$ possède la paire de distributions désirées $(0.3333x + 0.6667x^3, x^5)$.

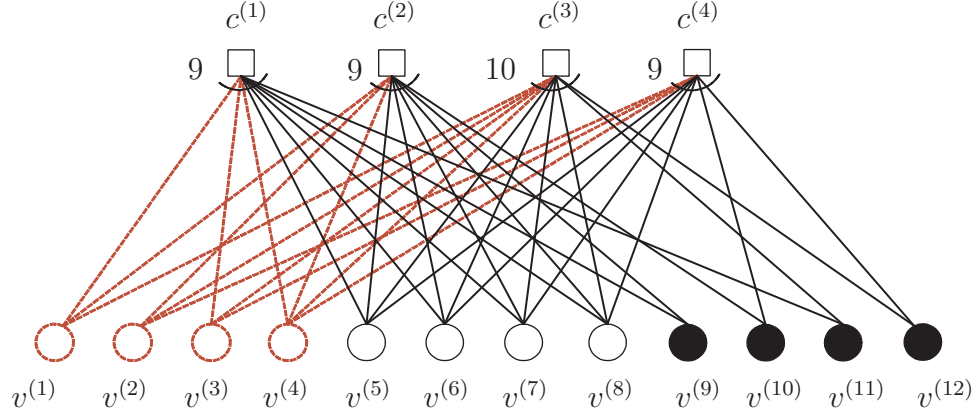


FIGURE 6.3 Protographe $G_{\mathcal{P}}$ d'un code RCDO mère de taux de codage 8/12 offrant le meilleur seuil de convergence asymptotique lorsque $d_{\lambda} = 4$.

L'observation précédente, nous montre qu'il existe des paires de distributions pour des codes RCDO mères qui offrent de bons seuils de convergence pour plusieurs taux de codage. Notons que l'exemple précédent fut obtenu par hasard et par conséquent, le lecteur intéressé pourrait continuer la recherche de protographes RCDO imbriqués qui possèdent de bon seuils de convergence théorique. Ceci permettrait la génération de codeurs RCDO mères qui offrent de bonnes performances d'erreur pour plusieurs taux de codage simultanément.

6.3 Décodeur itératif mère pour les codes RCDO imbriqués

Le décodage des codes RCDO imbriqués de taux de codage $(\frac{b-k}{c-k})$ s'effectue à partir du décodeur itératif mère utilisé pour décoder le code RCDO mère. Du point de vue du décodeur mère, c'est-à-dire le décodeur adapté au code RCDO mère de taux de codage b/c , à chaque instant i , le décodeur s'attend à recevoir c symboles provenant du canal. Or, pour un code RCDO imbriqué de taux de codage $(\frac{b-k}{c-k})$, seulement $(c - k)$ symboles sont générés par le codeur RCDO. Pour utiliser une seule structure de décodage pour tous les taux de codage imbriqués, il faut adapter le décodeur RCDO mère et attribuer des valeurs virtuelles aux k symboles absents à l'entrée du décodeur itératif RCDO mère. La valeur virtuelle choisie pour représenter

les symboles absents à l'entrée du décodeur itératif mère est la valeur binaire 0. Le décodeur itératif doit donc utiliser la métrique associée à la valeur binaire 0, c'est-à-dire la valeur $+\infty$, et l'attribuer aux symboles absents. La valeur absolue de cette métrique correspond à la fiabilité du symbole. La valeur infini correspond à la certitude absolue de l'information sur le symbole, et le signe de la métrique utilisée correspond à la polarité associée au bit 0 lors de la modulation BPSK. Cette métrique est utilisée pour représenter les messages initiaux provenant du canal ainsi que les messages qui correspondent à l'information extrinsèque des k symboles absents. Comme nous l'avons spécifié au Chapitre 3, le calcul de l'information extrinsèque s'obtient en deux étapes. La première étape consiste à transmettre les messages associés aux feuilles de l'arbre vers les noeuds de contraintes dans le graphe de Tanner des codes RCDO, et la seconde étape consiste à transmettre les messages des noeuds de contraintes vers les noeuds variables. La Figure 6.4 présente l'arbre de calculs $\mathcal{T}_{\mathcal{N}_2(v_i^{(m)})}$ utilisé à chaque

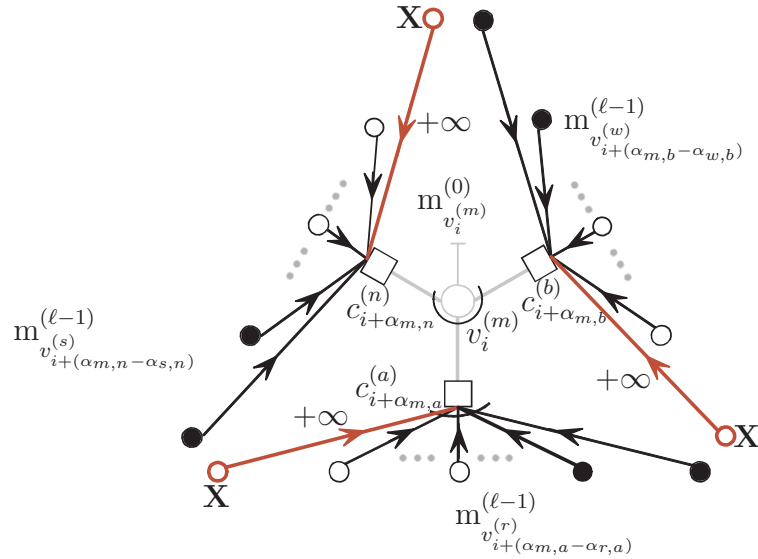


FIGURE 6.4 Sous-graphes induit $\mathcal{T}_{\mathcal{N}_2(v_i^{(m)})}$ de profondeur deux au voisinage du noeud variables v_i .

itération par le décodeur itératif mère pour calculer l'information extrinsèque associée au symbole $v_i^{(m)}$. Comme il est indiqué sur la figure, les messages propagés entre les symboles absents, identifiés par un X sur la figure, et les noeuds de contraintes, se voient attribués la valeur $+\infty$. Ceci représente la seule modification qui est requise dans l'implémentation du décodeur itératif mère. Notons que la valeur binaire 0 choisie pour représenter les symboles absents se justifie par le fait que les symboles absents

de l'équation (6.1) peuvent se voir virtuellement attribué la valeur binaire 0, car en attribuant cette valeur à ces symboles, le résultat de l'équation (6.1) demeure le même.

Toutefois, un désavantage accompagne la proposition effectuée, et vient du fait que la complexité associée à l'implémentation du décodeur est égale à celle associée aux décodeurs itératifs des codes RCDO mères. C'est-à-dire que la complexité de décodage ne décroît pas lorsque nous diminuons le taux de codage du code RCDO mère. Ceci va à l'encontre des résultats théoriques présentés à la Figure 5.7, où le nombre d'opérations nécessaires pour calculer l'information extrinsèque par itérations décroît lorsque le taux de codage diminue.

6.4 Performances des codes RCDO imbriqués réguliers $(3, d_\rho)$

Dans cette section, nous présentons les performances d'erreur obtenues par un code RCDO mère de taux de codage 16/20 qui possède un protographe régulier $(3, 15)$. Nous présentons aussi les performances d'erreur obtenues à partir des codes RCDO imbriqués au code mère qui permet d'atteindre les taux de codage 12/16, 8/12 et 4/8. Les résultats de simulations ont tous été obtenus en utilisant le décodeur itératif du code RCDO mère de taux de codage 16/20.

À la Figure 6.5, nous présentons le protographe $G_{\mathcal{P}}$ qui décrit le code RCDO mère utilisé dans cette section. Ce protographe est composé de 60 arêtes qui relient 20 noeuds variables à 4 noeuds de contraintes. Chaque noeud variable correspond à l'une des sorties du codeur RCDO mère. Sur la figure, les 16 sorties du codeur qui correspondent aux 16 entrées sont représentées par des cercles vides tandis que les 4 sorties du codeur qui correspondent aux 4 symboles de parité générés sont représentées par des cercles pleins. Nous pouvons remarquer que ce protographe est régulier, car tous les noeuds variables et tous les noeuds de contraintes possèdent leur degré égal à 3 et 15 respectivement. La matrice de contrôle $\mathbf{H}^T(D)$ correspondant à ce code RCDO mère se retrouve à l'Annexe C. En se référant à cette matrice, qui définit le code RCDO mère, nous pouvons constater que la mémoire du plus grand registre à décalage m_s associée au codeur RCDO mère est égale à 700.

La particularité de ce protographe vient du fait que les codes RCDO imbriqués

sont obtenus en n'admettant aucun symbole sur les k premières entrées du codeur RCDO mère, où $k = 4, 8, 12$. Et par conséquent, les protographes imbriqués $\tilde{G}_{\mathcal{P}_k} = G_{\mathcal{P}} \setminus \{v^{(1)}, \dots, v^{(k)}\}$ sont tous réguliers $(3, d_{\rho})$, $d_{\rho} = 6, 9, 12$. Par exemple, le code RCDO imbriqué de taux de codage 12/16 est généré en n'admettant aucun symbole aux quatre premières entrées du codeur RCDO mère, pour lequel correspond le protographe $\tilde{G}_{\mathcal{P}_4}$ qui correspond au protographe de la Figure 6.5 pour lequel les noeuds $v^{(1)}, \dots, v^{(4)}$ sont éliminés. Ainsi le protographe $\tilde{G}_{\mathcal{P}_4}$ est régulier $(3, 12)$, car tous les noeuds variables et de contraintes possèdent un degré égal à 3 et 12 respectivement. De la même manière, le code RCDO imbriqué possédant le taux de codage 8/12 est obtenu en n'admettant aucun symbole aux 8 premières entrées du codeur RCDO mère. Le code RCDO résultant possède le protographe $\tilde{G}_{\mathcal{P}_8}$ qui est régulier $(3, 9)$. Finalement, nous pouvons aussi générer un code RCDO imbriqué de taux de codage 4/8 en n'admettant aucun symbole aux 12 premières entrées du codeur RCDO mère. Le code RCDO résultant possède alors le protographe $\tilde{G}_{\mathcal{P}_{12}}$ qui est régulier $(3, 6)$.

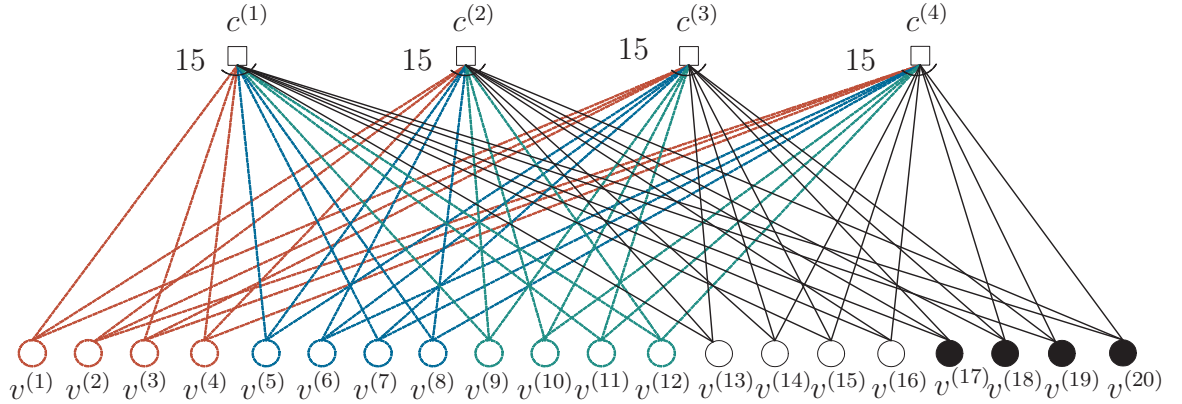


FIGURE 6.5 Protographe $G_{\mathcal{P}}$ régulier $(3, 15)$ d'un code RCDO mère de taux de codage 16/20.

À la Figure 6.6, nous présentons les performances d'erreur du code RCDO mère (Roy *et al.*, 2012), ainsi que celles obtenues par les codes RCDO imbriqués de taux de codage 12/16, 8/12 et 4/8. Comme nous l'indique cette figure, les performances d'erreur varient en fonction du taux de codage résultant, et au fur et à mesure que le taux de codage décroît meilleures deviennent les performances d'erreur des codes RCDO. Cette tendance s'explique facilement par le fait que pour un taux de codage faible, nous avons une proportion plus grande du nombre de symboles de parité qui

protègent les symboles d'information.

Nous avons aussi indiqué sur la figure les valeurs des seuils de convergence asymptotiques de l'algorithme de décodage qui sont associées aux différentes paires de distributions générées par les codes utilisés, c'est-à-dire $(3, d_\rho)$. Les résultats de simulations nous indiquent clairement que tous les codes RCDO imbriqués à partir du code RCDO mère permettent d'approcher la limite asymptotique qui est associée à leur paire de distributions.

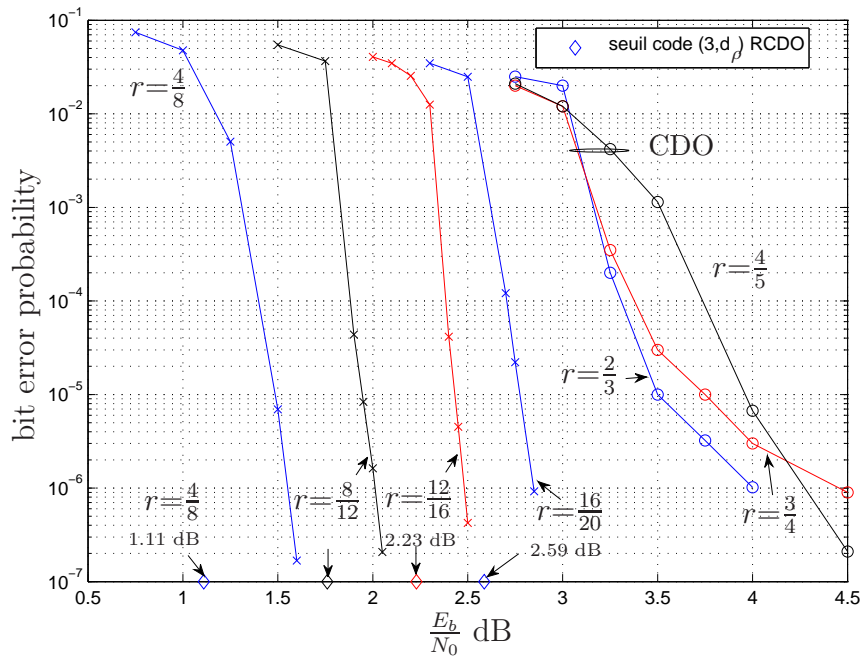


FIGURE 6.6 Performances d'erreur des codes imbriqués RCDO au code mère RCDO régulier $(3, 15)$ qui permet d'atteindre les taux de codage $4/8$, $8/12$, $12/16$, et $16/20$ à la suite de 25 itérations, $m_s = 700$. Aussi reproduit, les performances d'un code multi-taux de codage CDO avec $m_s = 74451$.

Remarquons qu'il y a un certain prix à payer lors du décodage en échange de la polyvalence du code RCDO mère. En effet au Tableau 6.1, nous présentons les différents paramètres associés au décodage d'une itération pour le code RCDO mère de taux de codage $16/20$. Nous présentons aussi dans ce tableau les paramètres du code 1 du Tableau 5.11 du chapitre 5. Ce code RCDO possède un taux de codage $4/8$ qui est identique au taux de codage obtenu en bloquant 12 entrées au codeur RCDO mère. Toutefois, en utilisant le décodeur itératif mère pour décoder le code

imbriqué, les paramètres associés au décodage itératifs demeurent les mêmes que ceux associés au code mère. Or, nous pouvons remarquer que pour le code RCDO 1, les paramètres reliés à la complexité sont largement inférieurs à ceux du code imbriqué qui possède le même taux de codage. Finalement, mentionnons que ce code RCDO mère

TABLEAU 6.1 Paramètres associés au décodage d'une itération du code RCDO mère (3,15) et du code 1 RCDO du Tableau 5.1.

code RCDO	nombre d'arêtes $ E_{\mathcal{P}} $	délai de décodage $m_s + 1$	unités de mémoire $(E_{\mathcal{P}} + c)(m_s + 1)$	nombre d'opérateurs \mathcal{R} $ E_{\mathcal{P}} $	nombre d'additions $ E_{\mathcal{P}} \lambda'(1)$
mère	60	701	56080	60	120
code 1	24	34	1088	24	48

régulier permet d'atteindre des performances d'erreur bien meilleures que toutes celles obtenues par les codes perforés doublement-orthogonaux présentés dans (Haccoun et Cardinal, 2005) et (Roy *et al.*, 2007). Nous présentons aussi à la Figure 6.6 les performances d'erreur d'un code CDO mère pouvant atteindre les taux de codage $2/3$, $3/4$ et $4/5$ provenant de (Haccoun et Cardinal, 2005). Ce code CDO possède une valeur m_s égale à 74 451. La première remarque que nous pouvons effectuer en comparant les deux codes mères est qu'un gain de codage supplémentaire d'au moins 1 dB peut être atteint pour chaque taux de codage désirés en utilisant le code RCDO mère plutôt que le code CDO mère. Cependant, ces performances d'erreur sont atteintes en effectuant cinq (5) fois plus d'itérations avec le code RCDO. Toutefois, notons que le code RCDO induit une latence bien moins élevée par itération que le code CDO. En effet, le ratio entre la valeur $(m_s + 1)$ du code CDO et la valeur $(m_s + 1)$ du code RCDO nous indique qu'il est possible d'effectuer 106 itérations avec le code RCDO pendant qu'une seule itération eût été effectuée pour le code CDO. En terme de latence totale $\mathcal{L}(m_s + 1)$, le code RCDO induit un retard de 17525 unités de temps contre 372225 unités de temps pour le code CDO, après avoir effectué 25 et 5 itérations respectivement. Le code RCDO mère offre donc une réduction de plus de 95% de la latence totale, et ce, tout en offrant des performances d'erreur largement supérieure à celles obtenues avec le code CDO mère.

6.5 Conclusion

Dans ce chapitre, nous avons introduit les codes RCDO imbriqués. Ces codes sont obtenus en n'admettant aucun symbole sur k entrées d'un codeur RCDO mère de taux de codage b/c générant ainsi des codes RCDO imbriqués de taux de codage $(\frac{b-k}{c-k})$, $1 \leq k < b$. Les différents taux de codage obtenus nous permettent donc d'envisager l'utilisation des codes RCDO au sein de systèmes de communications pour lesquels il est avantageux de pouvoir changer le taux de codage du code correcteur d'erreur. Nous avons aussi présenté les matrices de contrôle $\tilde{\mathbf{H}}_k^T(D)$ ainsi que le protographe $\tilde{G}_{\mathcal{P}_k}$ des codes RCDO imbriqués. Ceci nous a permis de soulever la problématique reliée au choix du protographe mère associé au code RCDO mère. De plus, nous avons présenté le décodeur itératif mère à partir duquel il est possible de décoder l'ensemble des codes RCDO imbriqués. Nous avons aussi présenté les performances d'erreur d'un code RCDO mère régulier (3,15) de taux de codage 16/20 qui permet de générer trois codes RCDO imbriqués régulier de taux de codage 12/16, 8/12 et 4/8. Les résultats de simulation présentés dans ce chapitre sont très encourageants. D'une part, ils sont à peine à quelques dixièmes de décibel de leur limite asymptotique respective, et d'autre part, les performances d'erreur obtenues par le code RCDO mère et ses codes imbriqués offrent un gain de codage non négligeable de plus de 1 décibel par rapport aux meilleurs codes convolutionnels doublement orthogonaux perforés déjà existants, et ce, tout en réduisant de plus de 95% la latence totale de décodage introduite par ces derniers.

Chapitre 7

CONCLUSION

7.1 Synthèse des travaux

Les bases de cette thèse reposent essentiellement sur les liens effectués au second chapitre entre les codes RCDO et les codes linéaires en blocs LDPC. Pour ces deux familles de codes, les mots de codes sont obtenus à partir de leur matrice de contrôle qui est creuse. En effectuant ce lien entre les deux familles de codes correcteur d'erreurs, nous avons pu utiliser la théorie des codes LDPC et plus précisément les résultats des travaux de (Richardson *et al.*, 2001) pour rechercher des protographes adaptés aux codes RCDO. D'ailleurs, nous avons pu observer que le protographe des codes RCDO doit contenir des noeuds variables ayant un degré plus grand que 1 si nous désirons approcher la capacité de Shannon des canaux BEC et AWGN. Cette condition rend donc obligatoire la récursivité des codeurs convolutionnels considérés. D'ailleurs, c'est précisément cette condition qui explique pourquoi les codes CDO et S-CDO, qui possèdent un protographe avec des noeuds variables de degré égal à un, n'offrent pas des performances d'erreur qui approchent la capacité de Shannon.

Nous avons aussi montré que les conditions de la double orthogonalité permettent l'élimination des petits cycles dans le graphe biparti de Tanner des codes RCDO. Plus précisément, les conditions de la double orthogonalité éliminent les cycles de longueurs 4, 6 et 8 dans le graphe de Tanner des codes RCDO. L'élimination de ces cycles permet au décodeur itératif d'estimer les symboles d'information, sur deux itérations successives, à l'aide d'un ensemble d'équations de contraintes qui sont indépendantes les unes des autres, ce qui entraîne une amélioration des performances d'erreurs.

Nous avons aussi évalué la complexité de calculs ainsi que les différents paramètres reliés à la mise en oeuvre du décodeur itératif utilisé pour décoder les codes RCDO. Comme nous l'avons décrit, le décodeur itératif utilisé est constitué d'une cascade de plusieurs simples décodeurs à seuil identiques. Le nombre de décodeurs à seuil dans la cascade représente le nombre d'itérations effectuées lors du décodage. Comme

nous avons pu le montrer, la complexité de calculs liées à l'estimation des symboles varie en fonction du nombre d'arêtes qui constitue le protographe des codes RCDO. De plus, il s'avère que la latence lors du décodage ainsi que la mémoire nécessaire à l'implémentation matérielle d'une itération sont des quantités qui varient en fonction de la taille du plus grand registre à décalage qui compose les codeurs RCDO. Or, cette valeur varie implicitement en fonction des conditions de la double orthogonalité imposées à la position des connexions des codeurs RCDO. Par conséquent, dans le but de générer des codes RCDO qui nécessitent une faible complexité lors de l'encodage et du décodage, nous avons fait la recherche de codes RCDO pour lesquels la mémoire du plus grand registre fut minimisée. La recherche de ces codes nous a montré qu'il est possible de générer des codes RCDO qui offrent de meilleures performances d'erreur que les codes CDO non récursifs tout en offrant une complexité de calculs, par itération, inférieure à celle qui est associée aux codes CDO.

Nous nous sommes aussi intéressés à la recherche des protographes adaptés aux codes RCDO qui permettent théoriquement de faire tendre le seuil de convergence théorique de la probabilité d'erreur vers la capacité de Shannon. Nous avons donc énoncé les meilleures paires de distribution qui correspondent aux codes RCDO pour les taux de codage $1/3$, $1/2$ et $2/3$. Les seuils furent calculés numériquement à l'aide de l'algorithme Évolution de la Densité de Probabilité. Les résultats obtenus pour les canaux BEC et AWGN, nous montrent qu'à ce jour, les protographes adaptés aux codes RCDO offrent des seuils de convergence théorique qui s'approche très près des résultats obtenus par les codes en blocs LDPC. Par exemple pour le canal AWGN, à peine un dixième de décibel favorise les seuils des codes LDPC au seuil des protographes des codes RCDO. En revanche, de façon pratique, les codes RCDO permettent facilement l'élimination des petits cycles, ce qui n'est pas nécessairement le cas pour les codes linéaires en blocs LDPC, où nous devons vérifier l'absence de petits cycles pour un graphe pouvant comporter plusieurs milliers de noeuds variables. Nous avons aussi présenté les performances d'erreur des codes RCDO dont le degré des noeuds variables (contraintes) des protographes RCDO fut optimisé. Nos résultats nous indiquent que tout comme pour les codes LDPC en blocs, les codes RCDO dont le protographe est irrégulier permettent d'approcher la capacité de Shannon pour les différents canaux de communications que nous avons considéré.

Finalement, nous avons défini les codes RCDO imbriqués. Ces codes sont obtenus à partir d'un code RCDO mère qui possède la particularité de pouvoir atteindre

plusieurs taux de codage. Comme nous l'avons présenté, la recherche de ces codes n'est pas un problème simple. Nous avons donc exposé les problématiques qui caractérisent ces codes. Ce qui nous a conduits à la recherche d'un code RCDO mère qui permet d'atteindre plusieurs taux de codage à partir de son protographe régulier. Les résultats de simulations nous ont montré que ces codes offrent de très bonnes performances d'erreur pour tous les taux de codage ciblés lors de la construction du code RCDO mère. De plus, il s'avère que ce seul code RCDO mère offre des performances d'erreur bien meilleures que celles générées, jusqu'à présent, par tous les codes perforés non récursif S-CDO qui offrent, eux aussi, plusieurs taux de codage.

En résumé, les codes RCDO présentés dans cette thèse n'ont rien à envier en terme de performances d'erreur aux codes convolutionnels Turbo et aux codes linéaires en blocs LDPC. Les performances d'erreur des codes RCDO varient en fonction de la structure du protographe associé aux codes RCDO ainsi qu'en fonction des conditions de double orthogonalité. En termes d'encodage, à l'encontre des codes Turbo, les codeurs RCDO ne nécessitent pas d'entrelaceur. Cependant, les codeurs convolutionnels utilisés dans la technique Turbo possèdent une longueur de contrainte très largement inférieure à celle des codes RCDO. Si l'on compare la complexité d'encodage des codes RCDO à celle des codes blocs LDPC, il est évident que la complexité de calculs nécessaires pour effectuer l'encodage des codes RCDO est inférieure à celle liée à l'encodage des codes LDPC. Car pour les codes RCDO, la complexité de calculs liée à l'encodage est proportionnelle au nombre d'arêtes qui composent le protographe du code RCDO tandis que la complexité de calculs liée à l'encodage des codes en blocs varient en fonction du nombre d'arêtes dans le graphe de Tanner des codes LDPC. Lorsque nous comparons la complexité liée au décodage itératif des codes RCDO à celles associées au décodage itératif des codes Turbo et LDPC. Nous pouvons constater qu'il est généralement plus complexe de décoder les codes Turbo car, l'algorithme BCJR utilisé pour décoder les codes Turbo est plus complexe à implémenter que l'algorithme à passage de messages utilisé pour décoder les codes RCDO et LDPC. En contrepartie, les codes Turbo nécessitent un bien moins grand nombre d'itérations en comparaison avec les codes RCDO et LDPC. Lorsque nous comparons le décodage des codes RCDO et LDPC, nous pouvons constater que pour atteindre une certaine probabilité d'erreur, les codes RCDO nécessitent un moins grand nombre d'itérations en comparaison avec les codes LDPC en bloc. Cependant, pour les codes RCDO le décodeur itératif contient autant de processeurs qu'il y a de décodeurs à seuil iden-

tiques mis en série. Il faut donc généralement plus de processeurs pour réaliser le décodeur itératif des codes RCDO qu'il n'en faut avec le décodeur itératif pour les codes LDPC. De plus, notons que la réalisation matérielle du décodeur itératif des codes RCDO est généralement plus simple que celle des codes en blocs LDPC, car un seul décodeur à seuil élémentaire doit être réalisé pour construire tout le décodeur itératif. En définitive, les codes RCDO et les codes LDPC en blocs représentent les deux facettes d'une même médaille.

7.2 Limitations de la solution proposée

Cette recherche comporte plusieurs limitations. Premièrement, notre étude survole uniquement les codes convolutionnels non terminés à temps invariant, c'est-à-dire que les connexions reliant les symboles à la sortie du codeur convolutionnels récurrents aux différents registres à décalage ne varient pas dans le temps. Ceci limite en quelque sorte l'ensemble des codes convolutionnels étudiés. Les auteurs de (Feldstrom et Zingirov, 1999) considèrent une classe encore beaucoup plus grande de codes convolutionnels, soit ceux à temps variant. En rendant variantes dans le temps la position des connexions du codeur convolutionnel, il devient alors possible de générer un code convolutionnel LDPC à partir d'une matrice de contrôle déjà existante d'un code en blocs LDPC. Cette méthode évite donc la recherche de matrices de contrôle.

Nous avons aussi volontairement limité notre recherche qu'aux codes convolutionnels obtenus à partir des trois conditions de la double orthogonalité. Nous n'avons pas cherché des conditions supplémentaires à imposer aux connexions, comme la triple orthogonalité, par exemple. Car, l'ajout de conditions supplémentaires aux codes conduira, selon notre expérience par rapport aux codes S-CDO et CDO, à augmenter la mémoire m_s du plus grand registre à décalage qui compose le codeur convolutionnel. Ceci entraîne une latence supplémentaire lors du décodage qui est souvent non négligeable.

Dans cette thèse, la modulation BPSK et une détection cohérente des signaux ont été utilisées pour générer nos résultats. Nous aurions pu envisager l'utilisation de modulations à efficacité spectrales plus élevées. Cependant, il aurait été difficile de comparer nos résultats. Car, pour des raisons historiques, la littérature est plus riche en termes de résultats lorsque la modulation BPSK est utilisée. Toutefois, si une modulation à efficacité spectrale élevée était envisagée alors il faudrait modi-

fier le décodeur itératif présenté pour que ce dernier puisse évaluer l'information extrinsèque pour chacun des bits qui compose le symbole transmis dans le canal de communication. Ceci aurait pour impact d'accroître la complexité de calculs liée au décodage.

De plus, nous avons limité notre recherche qu'aux canaux BEC et AWGN. Nous aurions pu envisager l'étude pour un canal à évanouissements. Cependant, nous pouvons justifier notre choix par le fait qu'il est bien connu que les codes correcteurs d'erreur qui offrent de bonnes performances d'erreur pour le canal AWGN offrent aussi de bonnes performances d'erreur pour les canaux à évanouissements lorsqu'un entrelaceur adéquat (désentrelaceur) est utilisé à la sortie du codeur de canal (à l'entrée du décodeur).

7.3 Améliorations futures

Dans cette section, nous attirons l'attention du lecteur vers d'autres pôles de recherche qui nécessiteraient d'être approfondis à partir des chapitres présentés dans cette thèse. D'un point de vue théorique, il serait intéressant maintenant de connaître les protographes adaptés aux codes RCDO, mais pour des canaux différents de ceux présentés dans cette thèse. Nous pourrions envisager la recherche de protographes adaptés aux codes RCDO, mais pour des canaux à évanouissements par exemple. Pour ce faire il faudrait utiliser l'algorithme de l'évolution de la densité de la probabilité présenté selon la méthode décrite dans (Hou *et al.*, 2001).

Il serait aussi intéressant de chercher des protographes RCDO, qui permettent de minimiser certains paramètres liés à la complexité lors du décodage. Dans cette thèse, nous avons minimiser la longueur du plus grand registre à décalage qui constitue les codeurs RCDO. Cette recherche nous a permis de minimiser la mémoire nécessaire à la réalisation des décodeurs itératifs ce qui se traduit en une réduction de la latence totale lors du décodage. Or, nous pourrions aussi évaluer le compromis entre la minimisation de la complexité de calculs et la variation du seuil théorique de convergence des codes RCDO.

Nous pourrions aussi considérer l'encodage et le décodage de symboles non pas binaires, mais plutôt \mathcal{A} -aire et ainsi généraliser la méthode proposée dans (Cardinal et Mvutu, 2009) pour les codes convolutionnels multiregistres RCDO. Ce type d'encodage (décodage) serait approprié lorsque nous envisageons l'utilisation de techniques

de modulations à efficacité spectrale plus élevée que celle de la modulation BPSK.

De plus, comme nous l'avons mentionné à la section précédente, nous nous sommes seulement intéressés aux codes RCDO non terminés. Or, selon nous, il devient primordial d'envisager le problème lié à la séquence de terminaison associée à un code convolutionnel RCDO. Cette avenue de recherche permettrait d'envisager l'utilisation des codes RCDO au sein de normes comme WiMax et où LTE. Ce problème est un problème ouvert en général pour tous les types de codes convolutionnels récurrents et consiste à insérer une séquence de bits à l'entrée d'un codeur RCDO de sorte que tous les registres du codeur RCDO reviennent à l'état initial, c'est-à-dire que tous les éléments de délai composant les registres à décalage du codeur possèdent la valeur zéro. D'une part, la terminaison des codes RCDO permettrait d'effectuer un décodage itératif plus efficace. En effet, il deviendrait possible d'utiliser une version itérative de l'algorithme de décodage BCJR (Bahl *et al.*, 1976) ce qui permettrait le décodage d'un code RCDO à partir des deux extrémités qui forment le graphe de Tanner. D'autre part, la terminaison des codes RCDO pourrait aussi offrir comme avantage l'encodage de paquets de données de tailles variables. Et par conséquent, les codes RCDO pourraient très bien être utilisés pour tous les types de transmissions par paquets comme les normes Ethernet, WiMax et LTE.

Références

- BAECHLER, B. (2000). *Génération de Codes Convolutionnels Doublement Orthogonaux*. M.sc.a., École Polytechnique de Montréal, Montréal.
- BAECHLER, B., HACCOUN, D. et GAGNON, F. (2000). On the search for self-doubly orthogonal codes. *Proceedings 2000 IEEE International Symposium on Information Theory*. Sorrento Palace Hotel, Sorrento, Italy, 292.
- BAHL, L. R., COCKE, J., JELINEK, F. et RAVIV, J. (1976). Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Transactions on Information Theory*, IT-20, 248–287.
- BERROU, C., GLAVIEUX, A. et THITIMAJSHIMA, P. (1993). Near shannon limit error-correcting coding and decoding : Turbo-codes. *ICC, 1993 Symposium*. Geneva, 1064–1070.
- CARDINAL, C. (2001). *Décodage à Seuil Itératif des Codes Convolutionnels Doublement Orthogonaux*. Thèse de doctorat, École Polytechnique de Montréal, Montréal.
- CARDINAL, C., HACCOUN, D. et GAGNON, F. (2003). Iterative threshold decoding without interleaving for convolutional self-doubly orthogonal codes. *IEEE Transactions on Communications*, IT-51, 1274–1282.
- CARDINAL, C., HACCOUN, D., GAGNON, F. et BATANI, N. (1998). Convolutional self-doubly orthogonal codes for iterative decoding without interleaving. *Proceedings 1998 IEEE International Symposium on Information Theory*. MIT, Cambridge, Mass. USA, 280.
- CARDINAL, C., HE, Y.-C. et HACCOUN, D. (2008). A new approach for the construction of powerful ldpc convolutional codes. *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*. 1176–1180.
- CARDINAL, C. et MVUTU, B.-F. (2009). Coded m-psk modulation using convolutional self-doubly orthogonal codes. *Vehicular Technology Conference, 2009. VTC Spring 2009. IEEE 69th*. 1 –5.
- CARDINAL, C., ROY, E. et HACCOUN, D. (2009). Simplified convolutional self-doubly orthogonal codes : search algorithms and codes determination. *Communications, IEEE Transactions on*, 57, 1674 –1682.

- CHEN, J. et FOSSORIER, M. (2002). Density evolution for two improved bp-based decoding algorithms of LDPC codes. *Communications Letters, IEEE*, 6, 208–210.
- CHUNG, S.-Y., FORNEY, G.D., J., RICHARDSON, T. et URBANKE, R. (2001). On the design of low-density parity-check codes within 0.0045 db of the Shannon limit. *Communications Letters, IEEE*, 5, 58–60.
- FELSTROM, A. J. et ZIGANGIROV, K. S. (1999). Time-varying periodic convolutional codes with low-density parity-check matrix. *IEEE Transactions on Information Theory*, IT-45, No 6, 2181–2191.
- GALLAGER, R. G. (1962). Low-density parity-check codes. *IRE Transactions of Information Theory*, 21–28.
- GALLAGER, R. G. (1968). *Information Theory and Reliable Communication*. Wiley and Sons, New York, NY.
- HACCOUN, D. et CARDINAL, C. (2005). High-rate punctured convolutional self-doubly orthogonal codes for iterative threshold decoding. *Communications, IEEE Transactions on*, 53, 55–63.
- HAGENAUER, J., OFFER, E. et PAPKE, L. (1996). Iterative decoding of binary block and convolutional codes. *IEEE Transactions on Information Theory*, IT-42, 429–445.
- HE, Y.-C., CARDINAL, C. et HACCOUN, D. (2008). Comparison of decoding complexities for LDPC and convolutional self-doubly-orthogonal codes. *Personal, Indoor and Mobile Radio Communications, 2008. PIMRC 2008. IEEE 19th International Symposium on*. 1–5.
- HE, Y.-C. et HACCOUN, D. (2005). An analysis of the orthogonality structures of convolutional codes for iterative decoding. *IEEE Transactions on Information Theory*, IT-51, 3247–3261.
- HE, Y.-C., HACCOUN, D. et CARDINAL, C. (2009). Error performances of multi shift-register convolutional self-doubly-orthogonal codes. *Communications Letters, IEEE*, 13, 685–687.
- HOU, J., SIEGEL, P. et MILSTEIN, L. (2001). Performance analysis and code optimization of low density parity-check codes on rayleigh fading channels. *Selected Areas in Communications, IEEE Journal on*, 19, 924–934.
- KIM, N. et PARK, H. (2004). Modified ump-bp decoding algorithm based on mean square error. *Electronics Letters*, 40, 816–817.

- KSCHISCHANG, F., FREY, B. et LOELIGER, H.-A. (2001). Factor graphs and the sum-product algorithm. *Information Theory, IEEE Transactions on*, 47, 498–519.
- LIN, S. et COSTELLO, D. (2004). *Error Control Coding : Fundamentals and Applications*. Prentice Hall, Inc. Englewood Cliffs, New Jersey.
- LUBY, M., MITZENMACHER, M., SHOKROLLAHI, M. et SPIELMAN, D. (2001). Improved low-density parity-check codes using irregular graphs. *Information Theory, IEEE Transactions on*, 47, 585–598.
- LUBY, M., MITZENMACHER, M., SHOKROLLAHI, M.A., S. D. et V., S. (1997). Practical loss-resilient codes. *Symp. on Theory of Computing*, 150–159.
- LUBY, M., SHOKROLLAHI, A., MITZENMACHER et SPIELMAN, D. (1998). Improved low-density parity-check codes using irregular graphs and belief propagation. *Information Theory, 1998. Proceedings. 1998 IEEE International Symposium on*, 117.
- MACKAY, D. J. C. (1999). Good error-correcting codes based on very sparse matrices. *IEEE, Transactions on Information Theory*, IT-45, 399–431.
- MASSEY, J. L. (1963). *Threshold Decoding*. MIT Press, Cambridge, MA.
- NEMR, A., CARDINAL, C., SAWAN, M. et HACCOUN, D. (2008). Very high throughput iterative threshold decoder for convolutional self-doubly orthogonal codes. *Circuits and Systems and TAISA Conference, 2008. NEWCAS-TAISA 2008. 2008 Joint 6th International IEEE Northeast Workshop on*. 257–260.
- ORLITSKY, A., VISWANATHAN, K. et ZHANG, J. (2005). Stopping set distribution of LDPC code ensembles. *Information Theory, IEEE Transactions on*, 51, 929–953.
- PUSANE, A., SMARANDACHE, R., VONTOBEL, P. et COSTELLO, D. (2011). Deriving good LDPC convolutional codes from ldpc block codes. *Information Theory, IEEE Transactions on*, 57, 835–857.
- RICHARDSON, T., SHOKROLLAHI, M. et URBANKE, R. (2001). Design of capacity-approaching irregular low-density parity-check codes. *Information Theory, IEEE Transactions on*, 47, 619–637.
- RICHARDSON, T. et URBANKE, R. (2001a). The capacity of low-density parity-check codes under message-passing decoding. *Information Theory, IEEE Transactions on*, 47, 599–618.

- RICHARDSON, T. et URBANKE, R. (2001b). Efficient encoding of low-density parity-check codes. *Information Theory, IEEE Transactions on*, 47, 638–656.
- RICHARDSON, T. et URBANKE, R. (2008). *Modern Coding Theory*. Cambridge University Press, Cambridge.
- RICHTER, G. et HOF, A. (2006). On a construction method of irregular LDPC codes without small stopping sets. *Communications, 2006. ICC '06. IEEE International Conference on*. vol. 3, 1119–1124.
- ROY, E., CARDINAL, C. et HACCOUN, D. (2007). Simplified high-rate punctured convolutional self-doubly orthogonal codes. *Proceedings 2007 IEEE International Symposium on Information Theory*. Nice, France, 2696–2699.
- ROY, E., CARDINAL, C. et HACCOUN, D. (2010). Recursive convolutional codes for time-invariant LDPC convolutional codes. *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*. 834–838.
- ROY, E., CARDINAL, C. et HACCOUN, D. (2012). A new construction of doubly-orthogonal regular ldpc codes and performances evaluation, soumis nov. 2012. *Information Theory, IEEE Transactions on*.
- SHANNON, E. C. (1948). A mathematical theory of communication. *Bell Syst. Tech. J.*, 27, 379–423, 623–656.
- SHARON, E. et LITSYN, S. (2006). A method for constructing LDPC codes with low error floor. *Information Theory, 2006 IEEE International Symposium on*, 2569–2573.
- TANNER, R., SRIDHARA, D., SRIDHARAN, A., FUJA, T. et COSTELLO, D.J., J. (2004). LDPC block and convolutional codes based on circulant matrices. *Information Theory, IEEE Transactions on*, 50, 2966–2984.
- TANNER, R. M. (1981). A recursive approach to low complexity codes. *IEEE Transactions on Information Theory*, Vol. 27 No. 5, 533–547.
- THORPE, J. (2003). Low density parity check (LDPC) codes constructed from protographs. *JPL INP Progress Report 42-154*.
- WIBERG, N. (1996). *Codes and decoding on general graphs*. Thèse de doctorat, Linköping University, Linköping, Sweden.

Annexe A

Conditions de la double orthogonalité des codes RCDO

A.1 Conditions de la double orthogonalité des codes RCDO

Dans cette annexe, nous présentons les conditions de la simple ainsi que celles de la double orthogonalité des codes convolutionnels multiregistres LDPC de taux de codage $r = b/c$. À partir de la définition 3.4 du Chapitre 3,

Définition 3.4 : Un codeur convolutionnel récursif multiregistres LDPC de taux de codage $r = b/c$ est *simplement orthogonal* si la position des connexions du codeur est telle que : les différences $(\alpha_{k,n} - \alpha_{k,m})$ sont distinctes des différences $(\alpha_{s,n} - \alpha_{s,m})$, $k \neq s$, $m \neq n$, $k, s \in \{1, \dots, c\}$ et $m, n \in \{1, \dots, (c - b)\}$ et $\alpha_{k,m} \in \mathbb{N}$.

nous pouvons en tirer le théorème suivant.

Théorème 3.1 : Tous les codes convolutionnels récursifs multiregistres LDPC *simplement orthogonaux* possèdent un graphe biparti dont la longueur du plus petit cycle est supérieure à 4.

Preuve : Pour montrer ce théorème il suffit d'analyser les équations de contraintes qui font intervenir par exemple le k -ième symbole à la sortie du codeur convolutionnel, $v_i^{(k)}$, $k \in \{1, \dots, c\}$. Étant donné que le code est récursif, ce symbole $v_i^{(k)}$ doit intervenir dans au moins 2 équations de contraintes. Par exemple, $c_{i+\alpha_{k,n}}^{(n)}$ et $c_{i+\alpha_{k,m}}^{(m)}$, $m \neq n$ avec $m, n \in \{1, \dots, (c - b)\}$. Ces deux équations de contraintes s'obtiennent à partir

de l'équation (3.2) et s'écrivent :

$$c_{i+\alpha_{k,n}}^{(n)} = v_i^{(k)} + \sum_{\substack{s=1 \\ s \neq k}}^c v_{i+\alpha_{k,n}-\alpha_{s,n}}^{(s)} \quad (\text{A.1})$$

et,

$$c_{i+\alpha_{k,m}}^{(m)} = v_i^{(k)} + \sum_{\substack{r=1 \\ r \neq k}}^c v_{i+\alpha_{k,m}-\alpha_{r,m}}^{(r)} \quad (\text{A.2})$$

Le sous-graphe induit par l'ensemble des variables qui décrivent (A.1) et (A.2) dans le graphe de Tanner d'un code convolutionnel récursif est présenté à la Figure A.1 a). Les feuilles de l'arbre représentent les variables qui apparaissent dans la somme des équations (A.1) et (A.2) et la racine de l'arbre est la variable commune aux 2 équations de contraintes soit $v_i^{(k)}$. Supposons qu'un cycle existe dans le sous-graphe induit, comme celui qui est présenté à la Figure A.1 b). Ceci veut dire qu'il existe une chaîne de sommets $v_{i+(\alpha_{k,n}-\alpha_{s,n})}^{(s)}, c_{i+\alpha_{k,n}}^{(n)}, v_i^{(k)}, c_{i+\alpha_{k,m}}^{(m)}, v_{i+(\alpha_{k,m}-\alpha_{r,m})}^{(r)}$ tel que les sommets aux extrémités de la chaîne représentent le même noeud, soit la même variable. Si tel est le cas, alors $r = s$, et

$$(i + \alpha_{k,n} - \alpha_{s,n}) = (i + \alpha_{k,m} - \alpha_{s,m})$$

et donc un cycle de longueur 4 existe si $\alpha_{k,n} - \alpha_{k,m} = \alpha_{s,n} - \alpha_{s,m}$, ce qui contredit la condition sur la position des connexions des codes convolutionnels récursifs *simplement orthogonaux*. Notons que la condition de la *simple orthogonalité* ne dépend pas de l'indice temporel i associé à la variable représentant la racine, $v_i^{(k)}$. Et par conséquent, il n'y a pas de cycle de longueur 4 si pour toutes les valeurs de k , c'est-à-dire pour chaque noeud variable du protographe, les différences $(\alpha_{k,n} - \alpha_{k,m})$ sont différentes de $(\alpha_{s,n} - \alpha_{s,m})$, $k \neq s$, $m \neq n$, $k, s \in \{1, \dots, c\}$ et $m, n \in \{1, \dots, (c-b)\}$ et $\alpha_{k,m} \in \mathbb{N}$.

À partir de la définition 3.5 de la *double orthogonalité* présentée au Chapitre 3,

Définition 3.5 : Un code convolutionnel récursif multiregistres LDPC de taux de codage $r = b/c$ est *doublement orthogonal* et donc RCDO, si la

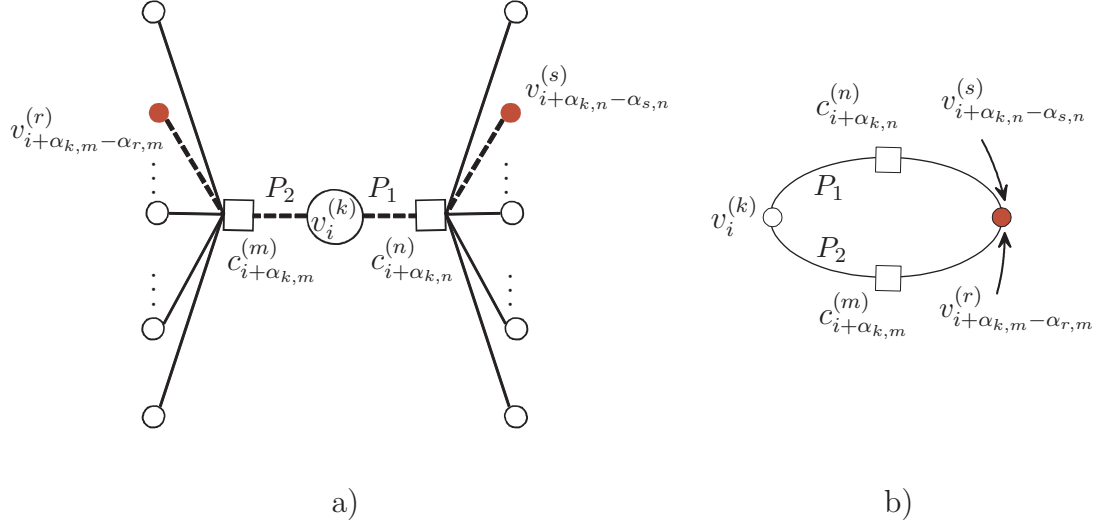


FIGURE A.1 a) Sous-graphe induit par les équations (A.1) et (A.2) dans le graphe de Tanner d'un code convolutionnel récursif simplement orthogonaux. b) Cycle de longueur 4.

position des connexions du codeur est telle que :

1. Les différences $(\alpha_{k,n} - \alpha_{k,m})$ sont distinctes des différences $(\alpha_{s,n} - \alpha_{s,m})$, $k \neq s$, $m \neq n$.
2. Les différences $(\alpha_{k,l} - \alpha_{k,q})$ sont distinctes des différences de différences $(\alpha_{s,l} - \alpha_{s,h}) - (\alpha_{r,h} - \alpha_{r,q})$, $k \neq r$, $k \neq s$, $l \neq q$, $h \neq q$, $s \neq r$, $h \neq l$.
3. Les différences de différences $(\alpha_{k,n} - \alpha_{k,q}) - (\alpha_{m,n} - \alpha_{m,p})$ sont distinctes des différences de différences $(\alpha_{f,p} - \alpha_{f,h}) - (\alpha_{r,q} - \alpha_{r,h})$, $k \neq m$, $k \neq r$, $f \neq m$, $f \neq r$, $h \neq p$, $h \neq q$, $p \neq n$, $q \neq n$.

où $k, p, r, s, t \in \{1, 2, \dots, c\}$ and $f, g, l, m, n \in \{1, 2, \dots, (c - b)\}$ et $\alpha_{k,n} \in \mathbb{N}$.

nous pouvons démontrer le théorème 3.2.

Théorème 3.2 : Tous les codes RCDO de taux de codage $r = b/c$ possèdent un graphe biparti dont la longueur du plus petit cycle est supérieure à 8.

Pour montrer le théorème, on suppose que le code convolutionnel récursif LDPC multiregistres est *a priori* simplement orthogonal et les connexions

$v_i^{(k)}$. Les noeuds de contraintes qui se situent à la profondeur 3 de l'arbre représentent les équations de contraintes pour lesquelles interviennent les variables représentées par les noeuds variables à la profondeur 2 et 4 de l'arbre. Par exemple, le noeud de contraintes $c_{i+(\alpha_{k,m}-\alpha_{r,m})+\alpha_{r,h}}^{(h)}$ de la Figure A.2 a) représente l'équation (A.3).

$$c_{i+(\alpha_{k,q}-\alpha_{r,q})+\alpha_{r,h}}^{(h)} = v_{i+(\alpha_{k,q}-\alpha_{r,q})}^{(r)} + \sum_{\substack{w=1 \\ w \neq r}}^c v_{i+(\alpha_{k,q}-\alpha_{r,q})-(\alpha_{w,h}-\alpha_{r,h})}^{(w)} \quad (\text{A.3})$$

Les cycles de longueur 6 peuvent être formés soit par une chaîne de longueur 2 et d'une chaîne de longueur 4 qui passent par la racine de l'arbre, ou bien par 2 chaînes de longueur 3, qui passent par le noeud $v_i^{(k)}$. Par exemple, les deux chaînes P_1 et P_2 de la Figure A.2 a) forment un cycle comme celui indiqué à la Figure A.2 b) si les noeuds aux extrémités des deux chaînes sont identiques, alors $w = s$, et

$$i + (\alpha_{k,l} - \alpha_{s,l}) = i + (\alpha_{k,q} - \alpha_{r,q}) - (\alpha_{s,h} - \alpha_{r,h})$$

ce qui contredit la deuxième condition de la Définition 3.5, où pour une valeur k les différences $(\alpha_{k,l} - \alpha_{s,l})$ doivent être distinctes des différences de différences $(\alpha_{k,q} - \alpha_{r,q}) - (\alpha_{s,h} - \alpha_{r,h})$ où $k \neq r$, $k \neq s$, $s \neq r$. Par conséquent, ce type de cycle de longueur 6 est évité dans le graphe de Tanner si pour tous les types- k de racine la condition 2 de la Définition 3.5 est vérifiée.

Notons qu'il est aussi possible d'avoir deux chaînes de longueurs 3 qui débutent à la racine de l'arbre, comme les chaînes P_3 et P_4 identifiées à la Figure A.2 a). Les 2 chaînes forment un cycle de longueur 6 si et seulement si les noeuds de contraintes aux extrémités des deux chaînes sont identiques, alors $s = p$ et donc,

$$i + (\alpha_{k,n} - \alpha_{m,n}) + \alpha_{m,p} = i + (\alpha_{k,q} - \alpha_{r,q}) + \alpha_{r,p}$$

cette dernière égalité est équivalente à,

$$(\alpha_{k,n} - \alpha_{k,q}) = (\alpha_{m,n} - \alpha_{m,p}) - (\alpha_{r,q} - \alpha_{r,p})$$

et contredit la deuxième condition de la Définition 3.5 où pour une valeur k les différences $(\alpha_{k,l} - \alpha_{k,q})$ doivent être distinctes des différences de différences $(\alpha_{s,l} - \alpha_{s,h}) - (\alpha_{r,h} - \alpha_{r,q})$. Notons que les 2 types de cycles de longueur 6 génèrent la même condition.

Jusqu'à présent, si la position des connexions des codeurs convolutionnels récursifs respectent les deux premières conditions de la Définition 3.5, le graphe de Tanner des codes est donc sans cycle de longueur 4 et 6. Pour éliminer les cycles de longueur 8, qui peuvent apparaître dans $\mathcal{T}_{\mathcal{N}_4(v_i^{(k)})}$, on doit vérifier que les chaînes de longueur 4, qui débutent à la racine de l'arbre ne possèdent pas la même extrémité. Un tel cycle est décrit à la Figure A.2 c) et existe si et seulement si les noeuds variables aux extrémités des 2 chaînes représentent le même noeud, c'est-à-dire que $w = f$, et

$$i + (\alpha_{k,n} - \alpha_{m,n}) - (\alpha_{f,p} - \alpha_{m,p}) = i + (\alpha_{k,q} - \alpha_{r,q}) - (\alpha_{f,h} - \alpha_{r,h})$$

avec $k \neq m$, $k \neq r$, $m \neq f$, $r \neq f$, $h \neq p$, $h \neq q$, $p \neq n$, $n \neq q$. Les cycles de longueurs 8 passant par la racine sont donc évités en imposant la troisième condition des codes convolutionnels récursifs doublement orthogonaux RCDO. Par conséquent, pour que le graphe de Tanner ne possèdent aucun cycle de longueur 8, on doit vérifier pour toutes les valeurs de k que les différences de différences soient distinctes des autres différences de différences. Un codeur RCDO possède un graphe de Tanner sans cycle de longueur 4, 6 et 8 si les trois conditions de la Définition 3.5 sont respectées.

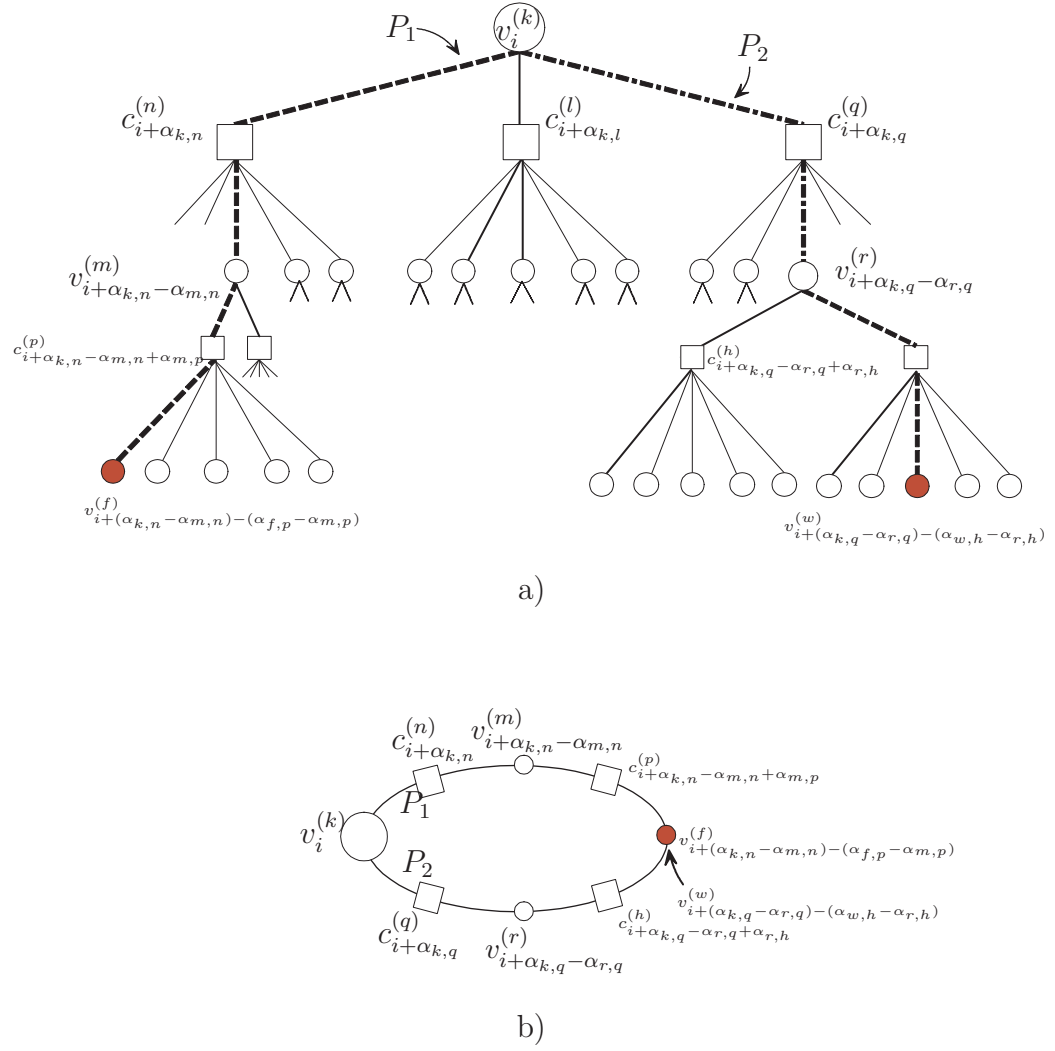


FIGURE A.3 a) Sous-graphe induit $\mathcal{T}_{\mathcal{N}_4(v_i^{(k)})}$ dans le graphe de Tanner d'un code convolutionnel RCDO. b) Cycle de longueur 6. c) Cycle de longueur 8.

Annexe B

Méthodes de génération des codes RCDO

B.1 Procédure de recherche des valeurs des connexions constituant la matrice de contrôle des codes RCDO

Dans cette annexe, nous présentons la méthode utilisée pour rechercher les valeurs $\alpha_{m,n}$ qui composent la matrice de contrôle $\mathbf{H}^T(D)$ des codes RCDO de taux de codage $r = b/c$, $m \in \{1, \dots, c\}$, $n \in \{1, \dots, (c-b)\}$. Nous rappelons que cette matrice est constituée de c lignes et de $(c-b)$ colonnes et s'écrit selon l'équation suivante.

$$\mathbf{H}^T(D) = \begin{pmatrix} h_{1,1}D^{\alpha_{1,1}} & \dots & h_{1,(c-b)}D^{\alpha_{1,(c-b)}} \\ h_{2,1}D^{\alpha_{2,1}} & \dots & h_{2,(c-b)}D^{\alpha_{2,(c-b)}} \\ \vdots & & \vdots \\ h_{b,1}D^{\alpha_{b,1}} & \dots & h_{b,(c-b)}D^{\alpha_{b,(c-b)}} \\ h_{b+1,1}D^{\alpha_{b+1,1}} & \dots & h_{b+1,(c-b)}D^{\alpha_{b+1,(c-b)}} \\ \vdots & & \vdots \\ h_{c,1}D^{\alpha_{c,1}} & \dots & h_{c,(c-b)}D^{\alpha_{c,(c-b)}} \end{pmatrix} \quad (\text{B.1})$$

La recherche des connexions $\alpha_{m,n}$ dans cette matrice s'effectue une fois après avoir généré la configuration des éléments non nuls $h_{m,n}$ qui vérifie la paire de distributions (λ, ρ) qui est associé au protographe du code RCDO. Notons que les éléments binaires non nuls $h_{m,n}$ ont été générés de façon aléatoire tout en s'assurant que les paires de distributions (λ, ρ) soient respectées.

Nous définissons S , $|E_{\mathcal{P}}|$ et m_s^{max} comme étant respectivement, le sous-ensemble des connexions qui vérifie les conditions de la double orthogonalité, le nombre d'éléments

non nul dans $\mathbf{H}^{\mathbf{T}}(D)$, et la valeur maximale que peut prendre la valeur des connexions $\alpha_{m,n} \in \mathbb{N}$. La génération d'un code s'effectue de la manière suivante : pour une valeur $m_s^{max} \in \mathbb{N}$ fixée, $S = \{\emptyset\}$:

1. Choisir un élément non nul dans la matrice de contrôle qui n'a pas déjà été choisi.
2. De façon aléatoire, générer une valeur entière $\alpha_{m,n}$ qui est inférieure à m_s^{max} et ajouter cette valeur au sous-ensemble des connexions S .
3. Si S vérifie les conditions de double orthogonalité aller à l'Étape 5, sinon aller à l'Étape 4.
4. Éliminer la nouvelle valeur ajoutée $\alpha_{m,n}$ de S et aller à l'Étape 2.
5. Si $|S| < |E_{\mathcal{P}}|$ aller à l'Étape 1, sinon arrêter, car nous avons trouvé une solution ;

Si après un certain temps nous ne trouvons pas de solution, nous recommençons une nouvelle recherche.

Annexe C

Matrices de contrôle des codes RCDO

Dans cette annexe, nous présentons les matrices de contrôle des codes convolutifs RCDO réguliers et irréguliers générés pour cette thèse.

C.1 Codes RCDO réguliers (3,6)

Dans cette section, nous présentons les matrices de contrôle des codes convolutionnels RCDO réguliers (3,6) présentés dans cette thèse. L'équation (C.1) représente la matrice de contrôle du code RCDO de taux de codage 15/30 ayant $m_s = 149$ dont les performances d'erreur sont présentées à la Figure 5.9.

$$\mathbf{H}^T(D) = \begin{pmatrix} 0 & 0 & 0 & D^{104} & 0 & 0 & 0 & 0 & 0 & D^{68} & 0 & D^{60} & 0 & 0 \\ 0 & 0 & 0 & 0 & D^{133} & 0 & 0 & 0 & 0 & D^{11} & 0 & 0 & D^{142} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & D^{62} & D^{29} & 0 & 0 & D^{84} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & D^{78} & 0 & 0 & 0 & D^{89} & 0 & D^{56} & 0 \\ 0 & 0 & D^{148} & 0 & 0 & 0 & 0 & 0 & 0 & D^{34} & D^{30} & 0 & 0 & 0 \\ 0 & D^{46} & D^{80} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & D^{112} & 0 \\ 0 & D^{57} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & D^{48} & D^{123} & 0 \\ 0 & D^{110} & 0 & 0 & 0 & 0 & 0 & D^{103} & 0 & D^{115} & 0 & 0 & 0 & 0 \\ D^{147} & 0 & 0 & D^{149} & 0 & D^{59} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & D^{20} & D^{119} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & D^{109} \\ 0 & 0 & D^{19} & 0 & 0 & 0 & 0 & 0 & D^{101} & 0 & 0 & 0 & 0 & D^{143} \\ 0 & 0 & D^{128} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & D^{139} & 0 & D^3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & D^{75} & 0 & 0 & 0 & 0 & D^{69} & D^{37} & 0 \\ 0 & D^{66} & 0 & 0 & 0 & D^{43} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & D^{114} \\ 0 & D^{15} & 0 & 0 & 0 & 0 & 0 & D^{88} & 0 & 0 & 0 & 0 & 0 & D^{50} \\ 1 & 0 & 0 & 0 & 0 & D^{33} & 0 & 0 & 0 & 0 & D^{108} & 0 & 0 & 0 \\ 0 & 1 & 0 & D^{114} & 0 & 0 & D^{138} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & D^{26} & 0 & 0 & D^{82} \\ 0 & 0 & 0 & 1 & 0 & D^{70} & 0 & D^7 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & D^{22} & 0 & 0 & 0 & 0 & 0 & D^{140} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & D^{77} & 0 & D^9 & 0 & 0 & 0 & 0 & 0 \\ D^{52} & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & D^{17} \\ 0 & 0 & 0 & 0 & 0 & D^{73} & 0 & 1 & D^{117} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & D^{14} & D^{73} & 0 & 0 & 1 & 0 & 0 & D & 0 & 0 \\ D^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & D^{116} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & D^{32} & 0 & 0 & 0 & 0 & 0 & D^{31} & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & D^{125} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & D^{45} \\ 0 & 0 & 0 & D^{111} & 0 & 0 & 0 & D^{10} & 0 & 0 & 0 & 0 & 1 & 0 \\ D^{121} & 0 & 0 & 0 & D^{105} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ D^{146} & 0 & 0 & 0 & 0 & 0 & 0 & D^{130} & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (\text{C.1})$$

Les Tableaux C.1 , C.2 et C.3 présentent les matrices de contrôle des codes RCDO réguliers (3, 6) dont les performances d'erreur sont présentées à la Figure 5.11.

TABLEAU C.1 Matrices de contrôle des codes RCDO réguliers (3, 6) de la Figure 5.11 ayant un taux de codage $r = b/c$ égal à $1/2$, où $b \in \{4, 5, 6\}$.

code RCDO	Matrice de contrôle $\mathbf{H}^T(D)$
code 1, $r = \frac{4}{8}$	$\begin{pmatrix} D^{33} & D^{18} & D^{11} & 0 \\ 0 & D^{14} & D^2 & D^{22} \\ D^{25} & D^{20} & 0 & D^2 \\ D^{10} & D^1 & 0 & D^{26} \\ 1 & D^{29} & D^{19} & 0 \\ 0 & 1 & D^{13} & D^{32} \\ D^{31} & 0 & 1 & D^5 \\ D^{15} & 0 & D^{17} & 1 \end{pmatrix}$
code 2, $r = \frac{5}{10}$	$\begin{pmatrix} 0 & 0 & D^{33} & D^{29} & D^4 \\ D^{33} & 0 & 0 & D^{24} & D^{13} \\ D^{22} & D & D^8 & 0 & 0 \\ D^{28} & 0 & D^{17} & D^9 & 0 \\ D^{30} & D^{15} & D^{11} & 0 & 0 \\ 1 & D^6 & 0 & 0 & D^{21} \\ 0 & 1 & 0 & D^{25} & D^{19} \\ 0 & D^3 & 1 & D^{26} & 0 \\ D^2 & 0 & 0 & 1 & D^{12} \\ 0 & D^{27} & D^5 & 0 & 1 \end{pmatrix}$
code 3, $r = \frac{6}{12}$	$\begin{pmatrix} 0 & D^{10} & 0 & 0 & D^{18} & D^{22} \\ 0 & 0 & D^3 & D^{19} & 0 & D^{29} \\ 0 & D & 0 & D^{22} & D^{16} & 0 \\ D^{34} & D^2 & 0 & D^6 & 0 & 0 \\ D^4 & 0 & 0 & 0 & D^{28} & D^{30} \\ 0 & 0 & 0 & D^{27} & D^7 & D^{28} \\ 1 & 0 & D^{17} & D^{15} & 0 & 0 \\ 0 & 1 & D^9 & 0 & 0 & D^{25} \\ D^{20} & 0 & 1 & 0 & D^{31} & 0 \\ D^{13} & D^{12} & 0 & 1 & 0 & 0 \\ 0 & D^{14} & D^{21} & 0 & 1 & 0 \\ D^8 & 0 & D^{32} & 0 & 0 & 1 \end{pmatrix}$

TABLEAU C.2 Matrices de contrôle des codes RCDO réguliers (3, 6) de la Figure 5.11 ayant un taux de codage $r = b/c$ égal à $1/2$, où $b \in \{7, 8\}$.

code RCDO	Matrice de contrôle $\mathbf{H}^T(D)$
code 4, $r = \frac{7}{14}$	$\begin{pmatrix} 0 & 0 & D^{15} & D^{18} & 0 & D^4 & 0 \\ 0 & D^{26} & D^3 & D^{25} & 0 & 0 & 0 \\ D^{34} & 0 & D^{11} & D^{20} & 0 & 0 & 0 \\ D^{16} & D^{21} & 0 & D^{13} & 0 & 0 & 0 \\ D^{28} & 0 & 0 & 0 & D^{23} & 0 & D^{14} \\ 0 & 0 & 0 & D^{19} & D^{12} & 0 & D^{22} \\ D^{29} & 0 & 0 & 0 & 0 & D^8 & D^{17} \\ 1 & 0 & D^{22} & 0 & 0 & D^{31} & 0 \\ D & 1 & 0 & 0 & 0 & 0 & D^{30} \\ 0 & D^7 & 1 & 0 & D^9 & 0 & 0 \\ 0 & 0 & 0 & 1 & D^5 & 0 & D^{24} \\ 0 & 0 & D^6 & 0 & 1 & D^{27} & 0 \\ 0 & D^{10} & 0 & 0 & D^{32} & 1 & 0 \\ 0 & D^2 & 0 & 0 & 0 & D^{30} & 1 \end{pmatrix}$
code 5, $r = \frac{8}{16}$	$\begin{pmatrix} 0 & 0 & D^{20} & D^{33} & 0 & D^{24} & 0 & 0 \\ 0 & 0 & D^{35} & D^6 & 0 & D^7 & 0 & 0 \\ 0 & 0 & 0 & D^{29} & D^{18} & 0 & 0 & D^{25} \\ D^{40} & D^{11} & 0 & 0 & D^9 & 0 & 0 & 0 \\ 0 & 0 & 0 & D^{21} & 0 & D & 0 & D^{13} \\ D^{23} & 0 & D^{31} & 0 & D^{26} & 0 & 0 & 0 \\ D^2 & 0 & D^{32} & 0 & D^{37} & 0 & 0 & 0 \\ 0 & D^{27} & D^{14} & 0 & 0 & D^3 & 0 & 0 \\ 1 & 0 & 0 & 0 & D^{36} & 0 & D^4 & 0 \\ 0 & 1 & 0 & D^{19} & 0 & 0 & D^{38} & 0 \\ D^{16} & 0 & 1 & 0 & 0 & D^5 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & D^{17} & D^{20} \\ D^8 & 0 & 0 & 0 & 1 & 0 & 0 & D^{19} \\ 0 & D^{30} & 0 & 0 & 0 & 1 & D^{10} & 0 \\ 0 & D^{28} & 0 & 0 & 0 & 0 & 1 & D^{22} \\ 0 & D^{15} & 0 & 0 & 0 & 0 & D^{12} & 1 \end{pmatrix}$

TABLEAU C.3 Matrices de contrôle des codes RCDO réguliers (3,6) de la Figure 5.11 ayant un taux de codage $r = b/c$ égal à $1/2$, où $b \in \{9, 10\}$.

code RCDO	Matrice de contrôle $\mathbf{H}^T(D)$
code 6, $r = \frac{9}{18}$	$\begin{pmatrix} 0 & 0 & 0 & D^{13} & D^{17} & D^{23} & 0 & 0 & 0 & 0 \\ 0 & 0 & D^{25} & 0 & 0 & D^{11} & 0 & 0 & D^{25} & 0 \\ 0 & D^{44} & 0 & 0 & D^{42} & D^{47} & 0 & 0 & 0 & 0 \\ 0 & D^{27} & D^{35} & 0 & D^{32} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & D^{12} & 0 & D^{22} & 0 & 0 & 0 & 0 & 0 \\ 0 & D^{24} & 0 & 0 & D^{10} & 0 & 0 & 0 & D^{38} & 0 \\ 0 & 0 & D^7 & 0 & 0 & D^{41} & 0 & D^{30} & 0 & 0 \\ 0 & D^9 & D^6 & 0 & 0 & D^{43} & 0 & 0 & 0 & 0 \\ D^{49} & D & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ D^{31} & 0 & 0 & D^{16} & 0 & 0 & 0 & 0 & D^8 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & D^{34} \\ 0 & 1 & 0 & D^{40} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & D^{46} & 0 & 0 & D^{28} & 0 & 0 & 0 \\ D^{18} & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ D^{33} & 0 & 0 & 0 & 1 & 0 & D^3 & 0 & 0 & 0 \\ 0 & 0 & 0 & D^2 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & D^{39} & D^4 & 0 \\ D^{14} & 0 & 0 & 0 & 0 & 0 & D^{19} & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & D^{21} & D^{36} & 1 & 1 \end{pmatrix}$
code 7, $r = \frac{10}{20}$	$\begin{pmatrix} 0 & 0 & 0 & D^{13} & D^{17} & D^{23} & 0 & 0 & 0 & 0 \\ 0 & 0 & D^{25} & 0 & 0 & D^{11} & 0 & 0 & D^{25} & 0 \\ 0 & D^{44} & 0 & 0 & D^{42} & D^{47} & 0 & 0 & 0 & 0 \\ 0 & D^{27} & D^{35} & 0 & D^{32} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & D^{12} & 0 & D^{22} & 0 & 0 & 0 & 0 & D^{15} \\ 0 & D^{24} & 0 & 0 & D^{10} & 0 & 0 & 0 & D^{38} & 0 \\ 0 & 0 & D^7 & 0 & 0 & D^{41} & 0 & D^{30} & 0 & 0 \\ 0 & D^9 & D^6 & 0 & 0 & D^{43} & 0 & 0 & 0 & 0 \\ D^{49} & D & 0 & 0 & 0 & 0 & 0 & 0 & 0 & D^{13} \\ D^{31} & 0 & 0 & D^{16} & 0 & 0 & 0 & 0 & D^8 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & D^{34} & D^{26} \\ 0 & 1 & 0 & D^{40} & 0 & 0 & 0 & 0 & D^{48} & 0 \\ 0 & 0 & 1 & D^{46} & 0 & 0 & D^{28} & 0 & 0 & 0 \\ D^{18} & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & D^{37} \\ D^{33} & 0 & 0 & 0 & 1 & 0 & D^3 & 0 & 0 & 0 \\ 0 & 0 & 0 & D^2 & 0 & 1 & 0 & 0 & 0 & D^{45} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & D^{39} & D^4 & 0 \\ D^{14} & 0 & 0 & 0 & 0 & 0 & D^{19} & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & D^{21} & D^{36} & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & D^{20} & D^{29} & 0 & 1 \end{pmatrix}$

La matrice de contrôle présentée à l'équation (C.2) est celle qui définit le code RCDO régulier (3, 6) de taux de codage 8/16, dont la valeur m_s est égale à 62. Les performances d'erreur de ce code RCDO sont présentées à la Figure 5.8 et 5.9.

$$\mathbf{H}^T(D) = \begin{pmatrix} 0 & D & 0 & 0 & D^{49} & 0 & 0 & 57 \\ D^{62} & 0 & D^{52} & D^{16} & 0 & 0 & 0 & 0 \\ D^{40} & D^{26} & 0 & 0 & 0 & D^{13} & 0 & 0 \\ D^{37} & D^{11} & 0 & 0 & 0 & D^2 & 0 & 0 \\ 0 & 0 & 0 & D^{21} & 0 & D^{61} & D^{28} & 0 \\ 0 & D^{44} & D^{45} & 0 & 0 & D^{41} & 0 & 0 \\ 0 & D^{22} & 0 & 0 & 0 & D^7 & D^5 & 0 \\ 0 & 0 & D^{35} & 0 & D^{46} & 0 & 0 & D^{17} \\ 1 & 0 & D^{10} & D^{38} & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & D^{50} & 0 & D^{20} & 0 \\ 0 & 0 & 1 & 0 & D^{23} & 0 & D^{54} & 0 \\ D^{24} & 0 & 0 & 1 & 0 & D^{58} & 0 & 0 \\ 0 & 0 & D^{59} & 0 & 1 & 0 & 0 & D^{47} \\ 0 & 0 & 0 & 0 & D^{31} & 1 & 0 & D^6 \\ 0 & 0 & 0 & D^{56} & 0 & 0 & 1 & D^{36} \\ D^8 & 0 & 0 & 0 & 0 & 0 & D^{53} & 1 \end{pmatrix} \quad (\text{C.2})$$

C.2 Code RCDO mère de taux de codage $r = 16/20$

Ce code RCDO mère a la particularité de pouvoir générer un code RCDO imbriqué régulier (3,12) de taux de codage 12/16, lorsque les lignes 5, 6, 7 et 8 sont éliminées de la matrice $\mathbf{H}^T(D)$ du code mère. Nous pouvons aussi générer un code RCDO imbriqué régulier (3,9) de taux de codage 8/12, lorsque les lignes 5 à 11 sont éliminées de la matrice $\mathbf{H}^T(D)$ du code mère. Finalement, nous pouvons aussi générer un code RCDO de taux de codage 4/8 en éliminant les lignes 5 à 15 de la matrice de contrôle du code mère. Les performances d'erreur de ces codes sont présentées à la Figure 6.6 du chapitre 6.

$$\mathbf{H}^T(D) = \begin{pmatrix} D^{700} & D^{685} & D^{449} & 0 \\ 0 & D^{569} & D^{50} & D^{634} \\ D^{669} & 0 & D^{292} & D^{45} \\ D^{594} & D^{104} & 0 & D^{694} \\ D^{381} & D^{471} & D^{444} & 0 \\ 0 & D^{12} & D^{616} & D^{534} \\ D^{340} & 0 & D^{175} & D^{327} \\ D^{426} & D^{18} & 0 & D^{393} \\ D^{608} & D^{10} & D^{613} & 0 \\ 0 & D^{495} & D^{490} & D^{421} \\ D^{606} & 0 & D^{299} & D^{37} \\ D^{173} & D^{485} & 0 & D^{458} \\ D^{273} & D^{152} & D^{486} & 0 \\ 0 & D^{358} & D^{551} & D^{188} \\ D^{537} & 0 & D^4 & D^{576} \\ D^{147} & D^{249} & 0 & D^{686} \\ 1 & D^{386} & D^{584} & 0 \\ 0 & 1 & D^{281} & D^{593} \\ D^{643} & 0 & 1 & D^{605} \\ D^{367} & D^{631} & 0 & 1 \end{pmatrix} \quad (\text{C.3})$$

TABLEAU C.5 Matrice de contrôle du code convolutionnel RCDO irrégulier. de taux de codage $r = b/c=1/2$, présentés à la Figure 5.4, $b = 10$.

code RCDO	Matrice de contrôle $\mathbf{H}^T(D)$
$r = \frac{10}{20}, m_s=300$	$\begin{pmatrix} D^{300} & D^{208} & D^{174} & D^{244} & D^{59} & D^{175} & D^{285} & D^{226} & D^{183} & D^{50} \\ 0 & 0 & 0 & 0 & D^{198} & D^{125} & 0 & D^{201} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & D^{227} & 0 & D^{135} & D^{133} \\ D^{262} & 0 & D^{297} & 0 & D^{95} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & D^{231} & 0 & D^{268} & 0 & D^{56} \\ 0 & D^{192} & 0 & D^{79} & 0 & 0 & D^{248} & 0 & 0 & 0 \\ D^{90} & 0 & 0 & 0 & D^{115} & 0 & 0 & 0 & 0 & 0 \\ D^{35} & D^{197} & D^{45} & D^2 & 0 & D^{68} & D^{281} & D^{214} & D^{34} & D^{153} \\ 0 & D^{279} & D^{151} & D^{249} & 0 & 0 & D^{189} & D^{81} & D^{40} & 0 \\ 0 & 0 & 0 & 0 & D^{290} & D^{240} & 0 & 0 & 0 & D^{69} \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & D^{137} \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & D^{161} & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & D^{155} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & D^{141} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & D^{87} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & D^{164} & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & D^{237} & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & D^{159} & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & D^{19} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ D^{107} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$

Les équations (C.4) et (C.5) représentent, respectivement, les matrices de contrôle correspondant aux codes RCDO irréguliers présentés à la Figure 5.10 pour les taux de codage 1/3, 1/2.

L'équation (C.4) représente la matrice de contrôle du code convolutionnel RCDO de taux de codage 1/3.

$$\mathbf{H}^T(D) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & D^{25} & D^{71} & 0 & D^{229} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & D^{142} & D^{225} & 0 & D^{154} \\ 0 & 0 & 0 & 0 & D^{21} & D^{151} & 0 & 0 & 0 & D^{54} \\ D^{250} & D^{163} & D^{242} & D^{99} & D^{101} & D^{38} & D^{217} & D^{34} & D^{130} & D^{122} \\ 0 & D^{228} & D^{47} & D^{105} & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & D^{104} & 0 \\ 0 & 10 & 0 & D^{73} & 0 & 0 & 0 & 0 & 0 & 0 \\ D^{124} & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & D^{208} & 0 & 0 \\ D^{68} & 0 & D^{26} & D^{78} & 1 & D & 0 & 0 & D^{147} & D^{106} \\ 0 & D^{126} & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & D^{74} & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & D^{174} & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & D^{10} & D^{199} & 1 & 0 \\ 0 & 0 & 0 & 0 & D^{77} & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (\text{C.4})$$

L'équation (C.5) représente la matrice de contrôle du code convolutionnel RCDO

de taux de codage 1/2.

$$\mathbf{H}^T(D) = \begin{pmatrix} 0 & D^{220} & 0 & 0 & D^{448} & 0 & 0 & 0 & 0 & D^{187} & 0 & 0 \\ 0 & D^{315} & 0 & 0 & 0 & 0 & 0 & D^{432} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & D^{182} & 0 & D^{85} & D^{266} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & D^{342} & 0 & 0 & 0 & 0 & 0 & D^{99} & 0 \\ D^{450} & D^{161} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & D^{155} & D^{371} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & D^{384} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & D^{227} & 0 & 0 & 0 & 0 & D^{12} \\ 0 & 0 & 0 & D^{131} & 0 & 0 & D^{205} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & D^{203} & 0 & 0 & D^4 & 0 & 0 \\ D^{447} & 0 & 0 & 0 & D^{196} & 0 & 0 & 0 & D^{326} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & D^{299} & 0 & 0 & 0 & D^{56} & 0 & 0 & 0 \\ 0 & D^{419} & 0 & 0 & 0 & 0 & 0 & D^{346} & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & D^{423} & D^{256} & 0 & 0 & 0 & 0 \\ 0 & 1 & D^{217} & D^{212} & D^{305} & D^{121} & D^{65} & D^{369} & D^{178} & 0 & D^{415} & D^{328} \\ D^{75} & D^{325} & 1 & D^{429} & 0 & D^{111} & 0 & D^{291} & D^{89} & D^{72} & D^{323} & D^{444} \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & D^{290} & D^{190} \\ D^{226} & 0 & D^{254} & D^{288} & 1 & D^{434} & 0 & 0 & D^{25} & D^{433} & D^{129} & D^5 \\ 0 & 0 & D^{327} & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & D^{302} \\ D^{183} & 0 & D^{297} & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ D^{420} & 0 & D^{108} & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & D^{351} & 0 & 0 & 1 & D^{87} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & D^{313} & 0 & 0 & D^{38} & 1 & 0 & 0 \\ 0 & 0 & 0 & D^{304} & 0 & 0 & 0 & 0 & 0 & D^{177} & 1 & 0 \\ 0 & 0 & 0 & D^{422} & 0 & D^{306} & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (\text{C.5})$$